

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**“IMPLEMENTACIÓN DE UN ANALIZADOR DE ESPECTROS
BASADO EN LA TRANSFORMADA RÁPIDA DE FOURIER
(FFT)”**

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PRESENTADO POR:

OSCAR AUGUSTO TORRES MEJÍA

**PROMOCIÓN
1989 - I**

**LIMA – PERÚ
2006**

**“IMPLEMENTACIÓN DE UN ANALIZADOR DE ESPECTROS BASADO EN LA
TRANSFORMADA RÁPIDA DE FOURIER (FFT)”**

***Dedico este trabajo a:
Mi Dios, por la sabiduría, fortaleza y
guía en el sendero del bien,
Mi Esposa, por la inspiración para
seguir adelante hacia la meta
en mi carrera y en mi vida
Mi Madre y Hermanas, por el apoyo y aliento.***

SUMARIO

El presente trabajo, recoge aspectos de la Teoría del Procesamiento Digital de Señales, tomando los fundamentos de la Transformada Rápida de Fourier (FFT), y los orienta a la aplicación práctica, mediante el diseño e implementación de un analizador de espectros digital de bajo costo, desarrollado sobre la base del software aplicativo MATLAB.

La primera parte del documento ofrece una visión general de la base conceptual de la teoría de procesamiento digital de señales y luego se profundiza en el fundamento teórico de la transformada rápida de Fourier (FFT).

En la siguiente parte del documento, se aborda la metodología para la implementación del propio analizador de espectros digital basado en el aplicativo MATLAB.

Luego, y como fruto de la aplicación del desarrollo del instrumento, se presentan el análisis de los resultados de la aplicación con señales reales en el analizador de espectros. Las señales tratadas son de tres tipos: dos del tipo biológico (EEG y ECG), y una del tipo sísmica.

Finalmente, se presenta algunas conclusiones y aportes sobre la aplicación.

ÍNDICE

PRÓLOGO	1
PLANTEAMIENTO DE INGENIERÍA DEL PROBLEMA	
1.1 Descripción del Problema	3
1.2 Objetivos del Trabajo	3
1.3 Limitaciones del Trabajo	4
CAPÍTULO II	
BASE CONCEPTUAL DEL PROCESAMIENTO DE SEÑALES	
2.1. Introducción al concepto de señales	5
2.2. Procesamiento digital de señales (DSP)	11
2.3. Fundamentos matemáticos del procesamiento de señales	14
2.3.1. Series de Fourier	14
2.3.2. Transformada de Fourier de tiempo continuo	16
2.3.3. Transformada inversa de Fourier de tiempo continuo	17
2.3.4. Transformada de Fourier de tiempo discreto	18
2.3.5. Transformada discreta de Fourier inversa	26
2.3.6. Transformada rápida de Fourier (FFT)	27
2.3.7. Implementación de la transformada rápida de Fourier (algoritmos)	44

CAPÍTULO III**METODOLOGÍA PARA LA SOLUCIÓN DEL PROBLEMA:
IMPLEMENTACIÓN DEL ANALIZADOR DE ESPECTROS DIGITAL**

3.1. Alternativas de solución	46
3.2. Solución al problema	47
3.3. Concepción práctica del sistema	48
3.4. Características y especificaciones del analizador de espectros	49
3.5. Soporte software	51
3.5.1. Aspectos de programación del software base	51
3.5.2. Implementación de algoritmos – programa de calibración	51
3.5.3. Implementación de algoritmos – programa de aplicación	51

CAPÍTULO IV**ANÁLISIS Y PRESENTACIÓN DE RESULTADOS**

4.1. Resultados del analizador de espectros – módulo de calibración	54
4.2. Resultados del analizador de espectros – módulo de aplicación	57
4.2.1. Señales de electrocardiograma	57
4.2.2. Señales de electroencefalograma	64
4.2.3. Señales sismográficas	70

CONCLUSIONES Y APORTES	75
-------------------------------	----

ANEXOS

ANEXO A.- TEOREMA DE PARSEVAL	79
ANEXO B.- TEOREMA DEL MUESTREO	81
ANEXO C.- DESARROLLO DE SCRIPTS EN MATLAB	85
ANEXO D.- MANUAL DEL INSTRUMENTO: ANALIZADOR DE ESPECTROS DIGITAL	112

BIBLIOGRAFÍA	117
---------------------	-----

PRÓLOGO

El presente documento ha sido elaborado, como un aporte a la investigación en el procesamiento digital de señales. Particularmente orientado al análisis espectral de señales de baja frecuencia, donde se ha encontrado aplicación en las señales del tipo biológicas y sísmicas.

El tema está orientado a profundizar en el conocimiento y aplicación práctica de la Transformada Rápida de Fourier (FFT), como un método más efectivo para el cálculo computacional del paso de un muestreo de señal analógica y continua en el tiempo y con la discretización de la señal, llevándolo al dominio de frecuencia, para precisamente analizar el comportamiento de las componentes de señal respecto a la frecuencia. De esta forma, se intenta obtener un instrumento de bajo costo, pero con las prestaciones requeridas en un ambiente de laboratorio.

El capítulo I comprende una breve identificación del problema; es decir, el no disponer de un analizador de espectros digitales asequibles; luego se plantea el objetivo del trabajo y finalmente las limitaciones del mismo.

En el capítulo II se ofrece una visión general de la base conceptual de la teoría de procesamiento digital de señales y se profundiza en el fundamento teórico de la Transformada Rápida de Fourier (FFT).

El capítulo III trata sobre la metodología para la implementación del propio analizador de espectros digital basado en el aplicativo MATLAB.

El capítulo IV se presentan el análisis de los resultados de la aplicación de señales reales en el analizador de espectros. Las señales tratadas son de tres tipos: dos del tipo biológico (EEG y ECG), y una del tipo sísmica.

El capítulo V se presenta algunas conclusiones y aportes sobre la aplicación.

El presente documento sin embargo, no cubre una implementación física a nivel de hardware, sino que se limita a una simulación en PC. Para este fin, se desarrolla previamente los conceptos teóricos necesarios, con miras a determinar la forma como se deben aplicar las herramientas de la "Transformada Rápida de Fourier - FFT", utilizando como software de desarrollo el aplicativo de ingeniería MATLAB.

En cuanto al método empleado para la investigación del tema, se ha recurrido a la información bibliográfica existente sobre el procesamiento digital de señales de los más destacados autores sobre el tema, así como consultas a páginas Web en Internet de universidades extranjeras y otros centros de investigación.

CAPÍTULO I

PLANTEAMIENTO DE INGENIERÍA DEL PROBLEMA

1.1. Descripción del problema

Es conocido el alto costo de los instrumentos para el análisis de espectros, y especialmente para el análisis de señales de baja frecuencia, y de los que existen, tienen limitaciones en la amplitud de rangos, pues no son muy flexibles. Por lo expuesto, el problema radica falta de accesibilidad para este tipo de instrumentos en los laboratorios de investigación, además de no contar con la flexibilidad de los instrumentos que existen en el mercado.

1.2. Objetivos del trabajo

Se tienen tres principales objetivos: Primero, explotar la utilidad de la Transformada Rápida de Fourier (FFT), aplicando los principios de su metodología compacta al diseño de un Analizador de Espectros Digital. Segundo: en base al aplicativo de software de ingeniería MATLAB y con el juego de funciones de la FFT, implementar el instrumento Analizador de Espectros Digital. Y tercero, aplicar el instrumento al análisis de señales reales de baja frecuencia del tipo biológico (ECG y EEG) y señales sísmicas.

1.3. Evaluación del problema

Descrito el problema, es requerido diseñar un instrumento para el análisis de frecuencias de señales de mediana y baja frecuencia de bajo costo, que a su vez sea precisa y flexible. Las soluciones analógicas no resultan adecuadas ni económicas, por lo que se precisa determinar una técnica más óptima, misma que sería provista con la ayuda del procesamiento digital de las señales, y la aplicación de la transformada rápida de Fourier.

1.4. Limitaciones del trabajo

El trabajo no contempla la implementación física del proyecto en cuanto a hardware, sino el diseño y a la aplicación de algoritmos de simuladores basados en aplicativos en PC. Así mismo, el análisis de datos de las señales está basado en información real, pero almacenada digitalmente en formato electrónico en archivos de la PC. El sistema diseñado tampoco contempla la interfase de un sistema de adquisición.

1.5. Síntesis del trabajo

Se plantea primeramente la ingeniería del problema, y luego se ofrece una visión general de la base conceptual de la teoría del DSP y se profundiza en el fundamento teórico de la transformada rápida de Fourier (FFT).

Seguidamente, se aborda la metodología para la implementación del instrumento analizador de espectros digital, y como fruto de la aplicación del desarrollo del instrumento, se presentan el análisis de los resultados de la aplicación con señales reales en el analizador de espectros. Las señales tratadas son de tres tipos: dos del tipo biológico (EEG y ECG), y una del tipo sísmica. Finalmente, se presentan algunas conclusiones y aportes sobre la aplicación.

CAPÍTULO II

BASE CONCEPTUAL DEL PROCESAMIENTO DE SEÑALES

2.1. Introducción al concepto de señales

Antes de introducir el concepto de procesamiento digital de señales, es necesario revisar la base conceptual física y matemática aplicada a las mismas. De una manera sencilla, se describen a las señales como funciones de una o más variables independientes y contienen información acerca de la naturaleza o comportamiento de algún fenómeno; así, los sistemas responden a señales particulares produciendo otras señales. Por ejemplo, la señal de voz se representa de forma matemática por la presión acústica como una función del tiempo, y una imagen se representa como una brillantez con respecto a dos variables espaciales.

Las aplicaciones de los conceptos de señales y sistemas aparecen en muchas áreas de la ciencia y tecnología como: las telecomunicaciones, aeronáutica, diseño de circuitos electrónicos, acústica, sismología, exploración petrolera, análisis de suelos, sistema de generación y distribución de energía, procesamiento de voz, biomédica, etc., y a pesar de esa gran amplitud y diversidad de campos; sin embargo, las características de las señales que los representan son las mismas y se manejan, en última instancia, bajo un mismo marco teórico.

Para poder describir a las señales y entender que información pueden estas contener, se requieren de herramientas para su descripción matemática y su manipulación. Se introduce a continuación algunos de los tipos más importantes de señales y algunos métodos simples para describirlas. Según el tipo de señal, se podrá usar el método más adecuado para poder manipularla.

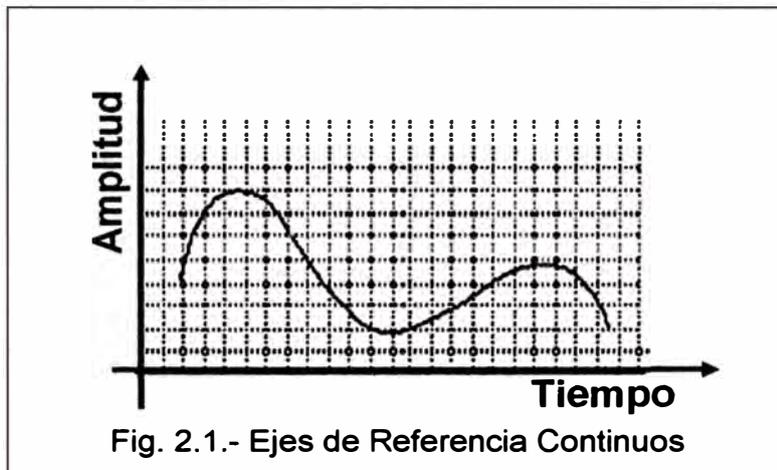
La señal más simple es la de una sola dimensión, o una sola variable independiente y lo que sigue es una clasificación de las mismas.

2.1.1 Señal Analógica:

Una señal analógica es una función continua de una variable también continua. Como se observa en la Fig. 2.1, esta se corresponde a los ejes abscisa y ordenada que son también continuos. Generalmente el eje de las abscisas se corresponderá con la variable t , que representa el tiempo, y el eje de las ordenadas a la magnitud de la señal analógica dada (1, 5).

En este contexto se define:

- **El rango de la Señal.-** Referido a la posible amplitud de valores que la señal puede tomar.
- **Eje de la Señal.-** El intervalo de tiempo para el cual la señal tiene existencia.



La frecuencia de una señal analógica es interpretada con la cantidad de períodos presentes de la señal por unidad de tiempo. Cuando se escoge la unidad en segundos, la frecuencia se mide en Hertz (Hz), donde:

- La frecuencia F
- El período de repetición de la onda T
- La relación: $F = 1/T$.

Como ejemplo de señal analógica, se tiene el voltaje o la corriente analógica producida; por ejemplo, por un micrófono o algún tipo de transductor (por ejemplo, como

el producido por el transductor de una ecosonda al recibir las señales del eco ultrasónico).

2.1.2. Señal Discreta en el Tiempo:

Una señal discreta en el tiempo, es una señal continua de una variable discreta. Referidos a la Fig. 2.2., el eje de las abscisas es discreto mientras que el eje de las ordenadas es continua. Para este tipo de señales, usualmente se asignan valores discretos de una variable n (es decir, toma como valor únicamente valores enteros positivos). En situaciones reales, las señales de tiempo discreto a menudo se originan como resultado de la aplicación de métodos de muestreo de señales analógicas¹ (1, 2, 3, 4, 5). La Fig. 2.2 muestra un ejemplo de la representación de señal discreta en el tiempo:

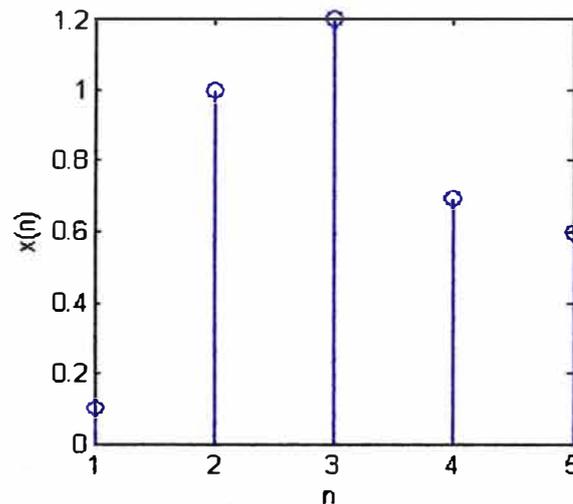


Fig. 2.2.- Señal de Tiempo Discreto

Notar que las señales son únicamente definidas para valores enteros a lo largo del eje de las abscisas, también referidos como puntos índices. Fuera de aquellos valores definidos, no existe ninguna información de la señal.

2.1.3. Señal Digital:

Sea una señal que es una función discreta de una variable discreta; es decir, si usáramos el plano de la Fig. 2.2, ahora los ejes de abscisas y ordenadas serían discretos, se dice entonces, que la señal representada es una señal digital. Como ejemplo de señales digitales se puede indicar la secuencia binaria, tal como la salida de

¹ Para más detalles, ver Anexo B sobre la teoría de Muestreo

un sistema de lectura de un disco compacto. Las señales digitales a menudo resultan del muestreo de una señal analógica y a cuyas muestras le son asignados a valores discretos (1, 2, 3, 4, 5).

La frecuencia digital es definida como:

$$f = \frac{F}{F_s} \quad (2.1)$$

Donde F_s es la frecuencia de muestreo. El intervalo de muestreo es la inversa de la frecuencia de muestreo:

$$T_s = \frac{1}{F_s} \quad (2.2)$$

Una señal discreta en el tiempo con frecuencia digital f por lo tanto tiene una frecuencia dada por:

$$F = f * F_s \quad (2.3)$$

2.1.4. Comparación de Señales periódicas y no periódicas:

Todas las señales a las que se hizo referencia más arriba, pueden ser periódicas. Para las señales de tiempo discreto y digitales se debe ser cuidadoso cuando se haga referencia a la periodicidad de la misma. La Fig. 2.3 muestra una señal periódica con período T_0 y una señal a-periódica.

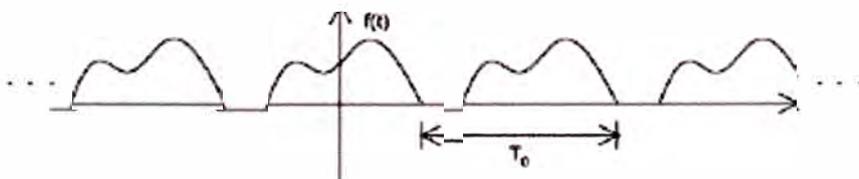


Fig. 2.3.A Señal Periódica

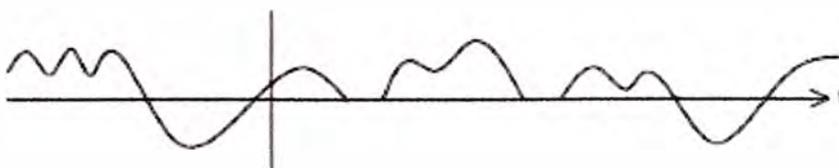


Fig. 2.3.B Señal a-periódica

2.1.5. Comparación de Señales Analógicas y Discretas:

Cuando se comparan las señales analógicas y las discretas en el tiempo, se encuentran varias similitudes. A menudo únicamente se substituye la variable t con n y la integración con una sumatoria. Sin embargo, hay diferencias importantes que es necesario precisar. Dado que la señal exponencial compleja es verdaderamente fundamental para el procesamiento de las señales, es necesario un mayor detalle.

- **Función Analógica.-**

La función exponencial compleja en escenario analógico es definida como:

$$x(t) = e^{j\Omega t} \quad (2.4)$$

Si Ω (rad/seg) es incrementada, la razón de la oscilación se incrementará continuamente. La función exponencial compleja es también periódica para algunos valores de Ω .

La figura 2.4 se grafica solo partes reales de la señal de $e^{j\pi t}$ y $e^{j3\pi t}$.

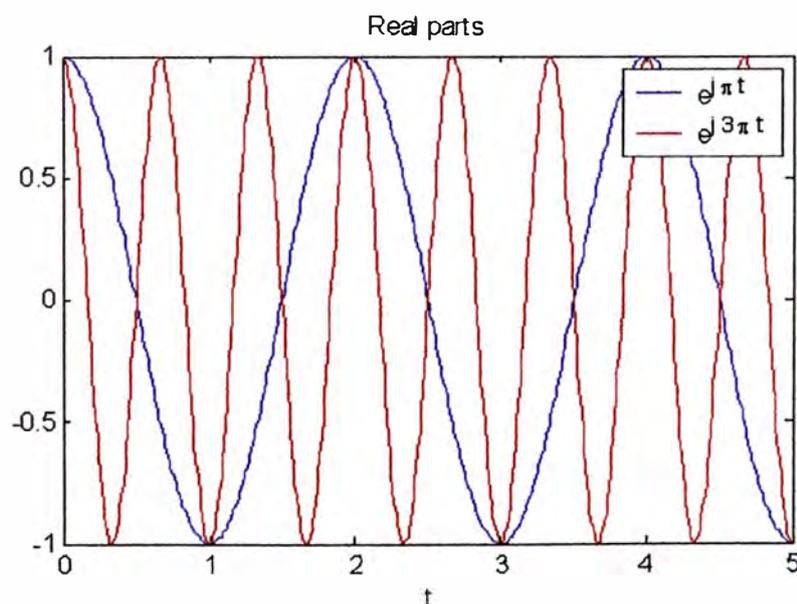


Fig. 2.4.- Partes reales de exponenciales complejas

- **Función Tiempo Discreto.-**

La función exponencial compleja para tiempo discreto es definida:

$$x[n] = e^{j\omega n} \quad (2.5)$$

Si ω (rad/muestra) es incrementada, la razón de la oscilación se incrementará y disminuirá periódicamente.

La razón es:

$$e^{j(\omega+2\pi k)n} = e^{j\omega n} + e^{j2\pi kn} = e^{j\omega n} \quad (2.6)$$

Donde: n, k , son valores enteros.

Esto implica que la exponencial compleja con frecuencia angular digital ω es idéntica a la exponencial compleja con $\omega_1 = \omega + 2\pi$: (Ver la Fig. 2.5)

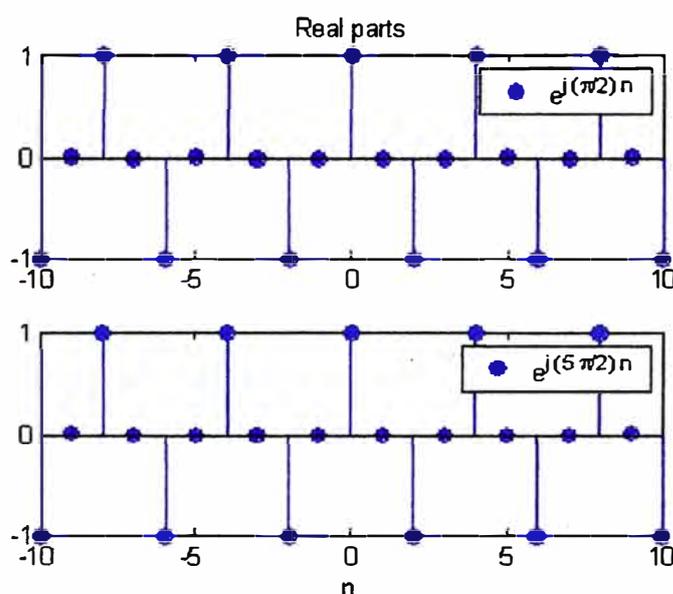


Figura 2.5.- Dos exponenciales que son idénticas en su representación

La razón de oscilación se incrementará hasta que $\omega = \pi$, entonces este se decrecerá y repetirá hasta 2π . En la Fig. 2.6 se aprecia que cuando se incrementa la frecuencia angular hasta π la razón de oscilación se incrementa.

Esto implica que la exponencial compleja con frecuencia angular digital ω es idéntica a la exponencial compleja con $\omega_1 = \omega + 2\pi$: (Ver la Fig. 2.6)

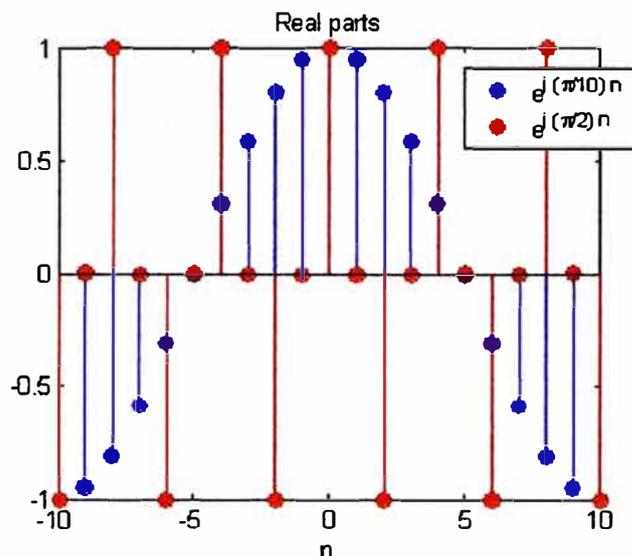


Fig. 2.6.- Dos exponenciales discretas con diferente frecuencia.

Como conclusión de ésta discusión, se debe considerar la exponencial de tiempo discreto solo en un (angular digital) intervalo de frecuencia 2π . Frecuencias bajas (angular digital) corresponderán a ω cerca a múltiplos pares de π . Frecuencias altas (angular digital) corresponderán a ω cerca a múltiplos impares de π .

2.2. Procesamiento digital de señales (DSP)

El procesamiento digital de señales (o de sus siglas en inglés DSP – Digital Signal Processing), como lo sugiere el término, comprende el procesamiento de señales por medios digitales. Una señal en este contexto puede referirse a un número diferentes de cosas, como ya lo explicamos en la sección previa, en general es un flujo de información representando cualquier cosa, desde los precios de la bolsa de valores, o las señales del electrocardiograma, o datos desde un sensor remoto en un satélite artificial. Históricamente el origen del procesamiento de señales proviene de la ingeniería eléctrica, y una señal aquí se refiere a la señal eléctrica portada por un cable, o tal vez por una onda de radio (4, 6).

2.2.1. Señales Analógicas y Digitales y el DSP

Para que las técnicas de procesamiento digital de señales puedan ser aplicadas a las señales analógicas, éstas primeramente deben ser convertidas en digitales; así, una

señal de voltaje analógico, podrá ser digitalizado a través de un dispositivo electrónico denominado convertidor analógico-digital (ADC), dando como resultado la generación de una salida digital en la forma de una secuencia binaria, representando valores numéricos, los cuales representan los voltajes de entrada del dispositivo.

2.2.2. Procesamiento de Señales

En los sistemas las señales comúnmente requieren ser procesadas, y hay una variedad de formas, según sea la necesidad y el entorno del sistema. Por ejemplo, tratándose de las señales generadas por los electrodos de un electrocardiograma aplicados a un paciente, se pueden producir contaminaciones en la señal original, produciendo ruidos, tales como los generados por algunos cambios pequeños de voltaje debido a la actividad de otros órganos y no necesariamente el corazón que es el objeto de observación; o también algunas alteraciones o interferencias de las fuentes de alimentación del sistema.

Por esto, antes del procesamiento de señales es necesario usar circuitos de filtrado que puedan remover o al menos reducir las señales indeseables que están afectando la señal principal. Con las nuevas técnicas de DSP se es capaz de filtrar las señales, mejorando la calidad de las mismas o para extraer parte importantes de la información contenida en ella, procedimientos que resultan mucho más ventajosos que las técnicas de la electrónica digital convencional.

2.2.3. Desarrollo del Procesador Digital de Señales

El desarrollo del procesador digital de señales, data desde los 1960's, con el uso de los grandes computadores digitales en las que se empezaron a implementar las aplicaciones numéricas como la proporcionada por la Transformada Rápida de Fourier (FFT), la cual permite se pueda calcular a alta velocidad el espectro en frecuencia de una señal.

Sin embargo, en aquellos tiempos, las técnicas del DSP no fueron ampliamente usadas, debido a que los grandes computadores solo estaban disponibles únicamente en las universidades y otras instituciones de investigación científicas.

La introducción del microprocesador a finales de los 1970's y comienzos de los 1980's hizo posible que sean usadas las técnicas DSP en un amplio rango de aplicaciones. Sin embargo, los microprocesadores de propósitos generales, tales como la familia Intel x86, no fueron los más ideales para los requerimientos de procesamiento numérico altamente intensivo de los DSP's; tal es así, que durante los 1980's la creciente importancia de los DSP condujo a los principales fabricantes (como Texas Instruments, Analog Devices y Motorola) a desarrollar Chips DSP – es decir, microprocesadores especializados, con la arquitectura diseñada especialmente para el tipo de operaciones requeridas en el procesamiento digital de señales.

En forma similar a un procesador de propósitos generales, un DSP es un dispositivo programable, con su propio juego de códigos de instrucción. Los Chips DSP son capaces de soportar millones de operaciones de punto flotante por segundo, y continuamente están siendo introducidas en el mercado versiones cada vez más rápidas y más poderosas de este tipo de dispositivos.

2.2.4. Aplicaciones del Procesador Digitales de Señales

Las principales aplicaciones actuales de la tecnología del DSP están en los dispositivos tales como los teléfonos móviles, computadora multimedia, grabadoras de video digital, reproductoras de CD, etc., reemplazando eficientemente la circuitería analógica de los teléfonos, televisores, video cámaras analógicas, etc.

Una aplicación importante de DSP está dada en los procesos de compresión y de-compresión de señales. Así, en los sistemas CD por ejemplo, la música grabada en el Disco Compacto, está en realidad, registrada en una forma comprimida (con lo cual se incrementa la capacidad de almacenamiento en dicha unidad), pero luego debe ser descomprimida para que la señal grabada pueda reproducirse, con un mínimo de error comparado con la señal original.

Este mismo tipo de compresión de señales, es también usada en los sistemas de telefonía celular para permitir un gran número de llamadas mantenidas en forma simultánea dentro de una celda local.

La tecnología de compresión de señales basada en el DSP, ha permitido no sólo hablar uno a otro vía teléfono, sino que se puedan tener comunicaciones multimedia de

una PC a PC o algún otro dispositivo multimedia, transmitiendo imágenes de las pequeñas cámaras de video usando tan solo una línea de teléfono convencional, gracias al uso eficiente del limitado ancho de banda del medio.

A pesar de la teoría matemática que soporta las técnicas del DSP tales como las Transformadas Rápida de Fourier y Hilbert, el diseño de filtros digitales y compresión de señales pueden ser moderadamente complejos, las operaciones requeridas para implementar estas técnicas son de hecho bastante simples, consistiendo principalmente de operaciones que pueden ser efectuadas por un chip calculador de cuatro funciones. La arquitectura de un chip DSP es diseñada para realizar operaciones a velocidades increíblemente altas, procesando hasta decenas de millones de muestras por segundo, con lo cual, provee un alto rendimiento en el procesamiento de señales en tiempo real; por ejemplo, para procesar una señal de imágenes en “vivo”, muestreándola y luego enviándola procesada a una pantalla de video digital. En realidad, muchas de las señales son en tiempo real, y requieren la aplicación de DSP en tiempo real.

2.3 Fundamentos Matemáticos del Procesamiento de Señales

2.3.1 Series de Fourier

Una de las herramientas fundamentales del análisis de frecuencia o espectral para señales periódicas en tiempo continuo, es dada por la representación matemática de la serie de Fourier (formulada por Jean Baptiste Fourier, matemático francés – 1768 - 1830). En general, una señal de forma periódica, $f(t)$, puede ser representada matemáticamente por la suma de un número infinito de términos senoidales y cosenoidales, afectados éstos por un término constante que representa la amplitud de cada uno de los componentes. En otras palabras, una señal puede ser representada por una suma ponderada de sinusoides relacionadas armónicamente, cuya representación general es dada por la fórmula (1):

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + \sum_{n=1}^{\infty} b_n \text{sen}(n\omega t) \quad (2.7)$$

Donde t es la variable independiente que a menudo representa el tiempo, pero en otros casos podría representar otras variables como por ejemplo la distancia. En nuestro caso, $f(t)$ representa típicamente la variación de la onda de voltaje respecto al tiempo, donde:

$$\omega = \frac{2\pi}{T_p} = 2\pi f \wedge f = \frac{1}{T_p} \quad (2.8)$$

Siendo ω la primera armónica o frecuencia angular fundamental, y relacionada con la frecuencia fundamental f , la cual es la inversa del período de la señal T_p .

El término a_0 es una constante igual al promedio de la señal $f(t)$ tomada durante un período, representando por ejemplo el nivel de voltaje continuo (componente DC) de la señal.

$$a_0 = \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} f(t) dt \quad (2.9)$$

Así mismo, a_n y b_n , representan las amplitudes de las componentes de la n -ésima armónica de ω , y $n=1, 2, 3, \dots, \infty$

$$a_n = \frac{2}{T_p} \int_{-T_p/2}^{T_p/2} f(t) \cos(n\omega t) dt \quad (2.10)$$

$$b_n = \frac{2}{T_p} \int_{-T_p/2}^{T_p/2} f(t) \text{sen}(n\omega t) dt \quad (2.11)$$

En una forma más compacta se puede expresar la serie de Fourier de la forma imaginaria:

$$f(t) = \sum_{n=-\infty}^{\infty} d_n e^{jn\omega t} \quad (2.12)$$

en la cual:

$$d_n = \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} f(t) e^{-jn\omega t} dt \quad (2.13)$$

Que es la notación compleja, donde además:

$$d_n = (a_n^2 + b_n^2)^{1/2} \wedge \phi_n = -\tan^{-1}\left(\frac{b_n}{a_n}\right) \quad (2.14)$$

Donde ϕ_n es la fase de la componente n -ésima armónica.

2.3.2. Transformada de Fourier de tiempo continuo

Muchas señales en la naturaleza; sin embargo, no son de carácter periódico, con lo cual, la serie de Fourier no puede aplicarse en estos casos; pero si acudimos a ciertos artificios, ésta puede ser modificada para que dicha señal no periódica sea representada matemáticamente en términos de senos y cosenos, convirtiendo la serie de Fourier en una integral de Fourier, denominada *Transformada de Fourier*, haciendo tender para esto T al infinito (1). En estas condiciones la fórmula $1/Tp=\omega/2\pi$ decrece a $d\omega/2\pi$ tendiendo a cero. Sin embargo, esto representa la otrora variable discreta $n*\omega$ que se convierte en una variable continua ω , por tanto, la amplitud y fase de la señal se convierte en una función continua de ω , de donde:

$$d(\omega) = \frac{d\omega}{2\pi} \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad (2.15)$$

Si se normaliza la fórmula dividiendo entre $d\omega/2\pi$ se obtiene:

$$F(j\omega) = \frac{d(\omega)}{\left(\frac{d\omega}{2\pi}\right)} = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad (2.16)$$

Donde el rango de las variables está definida para: $-\infty < t < \infty$ y $-\infty < \omega < \infty$

Las condiciones para poder obtener la transformada de Fourier son, las condiciones establecidas por Dirichlet:

- Que la señal sea absolutamente integrable, es decir:

$$\int_{-\infty}^{\infty} |f(t)|^2 dt < \infty \quad (2.17)$$

- Que tenga un grado de oscilación finito.
- Que tenga un número máximo de discontinuidades.

Por otro lado, la transformada de Fourier es una particularización de la transformada de Laplace con $S=j\omega$ (siendo $\omega=2\pi f$).

La integral resultante de Fourier $F(j\omega)$ es una función compleja y se la conoce con el nombre de Transformada de Fourier, y por ser compleja, se cumple que:

$$F(j\omega) = \text{Re}(j\omega) + j \text{Im}(j\omega) = |F(j\omega)|e^{j\phi(\omega)} \quad (2.18)$$

Entonces:

$$|F(j\omega)| = \left[\text{Re}^2(j\omega) + \text{Im}^2(j\omega) \right]^{1/2} \wedge \phi(\omega) = \tan^{-1} \left[\frac{\text{Im}(j\omega)}{\text{Re}(j\omega)} \right] \quad (2.19)$$

Las unidades de $F(j\omega)$, según 2.19 son *Volts/Hertz* (si la amplitud de la señal fuera en voltios) y representa la amplitud de la densidad espectral, y su fase asociada $f(j\omega)$.

Las transformadas lineales, especialmente las transformadas de Fourier y Laplace, son ampliamente usadas en ciencia e ingeniería. Por ejemplo, la Transformada de Fourier es usada en análisis de sistemas lineales, estudios sobre antenas, óptica, modelamiento de procesos aleatorios, teoría de las probabilidades, física cuántica, en la recuperación de datos astronómicos, etc. La versatilidad y amplitud de la transformada de Fourier, la convierte en una herramienta usada en muchos campos de la ciencia, matemática e ingeniería para hacer la resolución de los problemas una tarea más sencilla; así mismo, la Transformada de Laplace, siendo una generalización de la Transformada de Fourier para variables complejas, tiene también un amplio uso en aplicaciones de ingeniería.

2.3.3. Transformada inversa de Fourier de tiempo continuo

Con la transformada de Fourier se logra pasar del dominio del tiempo al dominio de la frecuencia de la señal. Así mismo, también es posible pasar del dominio de la frecuencia al dominio del tiempo de una señal, para lo cual se aplica la Transformada Inversa de Fourier, que se define de la siguiente manera:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega) e^{j\omega t} d\omega \quad (2.20)$$

Donde también se puede reemplazar $d\omega/2\pi$ por f , la frecuencia.

2.3.4. Transformada discreta de Fourier o de tiempo discreto

Dado que para el procesamiento moderno de señales se requiere cada vez más la intervención de los procesadores digitales de señal, y para esto el correspondiente muestreo o discretización de las señales, los mecanismos tradicionales de tratamiento continuo indicados en los puntos anteriores no son válidos. Se exige por tanto, de otros mecanismos que ayuden con el tratamiento de éstas señales digitales.

Por otro lado, se tiene que para una adecuada representación de la señal procesada digitalmente, ésta será lograda si se toman las muestras lo más juntas posibles; es decir, se tendrá una mayor precisión en la representación de la señal inicial cuanto mayor sea la frecuencia de muestreo aplicada a la misma. Pero de otro lado, debe establecerse la condición límite, pues una frecuencia de muestreo demasiado elevada, y que por tanto lleve a un número elevadísimo de muestras, podría conllevar a una imposibilidad en el procesamiento de la señal: sean por la capacidad limitada del procesador, como por la lentitud en reconstruir las señales procesadas en tiempo real, con lo cuál, carecería de sentido el discretizar la señal, cuando no se es capaz de representarla en el momento y a la velocidad requerida por un sistema. De esto, se deduce que debe establecerse un compromiso en los valores de la frecuencia de muestreo y la precisión con la que se desea representar la señal, esto ajustado a su vez, a la capacidad del sistema procesador de señales.

Además de todo esto, se tiene que todas estas señales reales que se desean discretizar son de duración finita, y no infinita ni continua como suele muchas veces representarse en los modelos matemáticos.

La solución matemática al problema del tratamiento de las señales por discretización que toma predominancia es la que contempla la teoría de Fourier para el caso discreto y finito. La Transformada Discreta de Fourier (DFT- Discrete Fourier Transform), algunas veces es llamada la transformada de Fourier finita y es ampliamente usada en el procesamiento de señales y campos relacionados al análisis de las frecuencias contenidas en una señal resultante de un muestreo o digitalización de la misma, interviniendo en una amplia variedad de algoritmos en los desarrollos para el DSP. En realidad, la DFT es una secuencia, no una función de una variable continua, y corresponde a muestras equiespaciadas en frecuencia de la transformada de Fourier de la señal (1, 2, 3, 4, 7).

a. Algunos conceptos de los dominios del tiempo y la frecuencia en relación al tratamiento de las Señales.-

Antes de profundizar en los conceptos de la transformada discreta de Fourier, es necesaria la revisión de algunos conceptos fundamentales relativos al entorno de las señales, dado que éstas pueden ser vistas desde dos puntos de vista diferentes (física y matemáticamente hablando), a los que denominaremos dominios:

a.1.- El dominio del tiempo.- El más usado es el dominio del tiempo, y para visualizar más fácilmente una señal bajo dicho dominio, se puede imaginar el trazo en un osciloscopio de una señal, donde la deflexión vertical en la pantalla representa la amplitud de la señal, y la deflexión horizontal, representa la variable tiempo. Por tanto, en el dominio del tiempo, la amplitud de la señal es una función de la variable independiente tiempo.

a.2.- El dominio de la frecuencia.- Una segunda forma de representar una señal es en el dominio de la frecuencia, y para visualizar más fácilmente una señal bajo este concepto, se puede imaginar el trazo en analizador de espectros, donde la deflexión horizontal en la pantalla de este instrumento es la variable frecuencia y la deflexión vertical es la amplitud de la señal a dicha frecuencia (o el contenido de potencia de dicha señal para cada valor de la frecuencia). Por tanto, en el dominio de la frecuencia, la amplitud de la señal es una función de la variable independiente frecuencia.

Se tiene que para cualquier señal, ésta puede ser completamente descrita (física y matemáticamente hablando) en cualquiera de estos dos dominios. Además, podemos pasar de un dominio a otro usando herramientas como las Transformadas: La Transformada de Fourier para pasar del dominio del tiempo al dominio de la frecuencia y la Transformada de Fourier Inversa para pasar del dominio de la frecuencia al dominio del tiempo.

En el tratamiento matemático de las señales, y dependiendo de lo que se desea hacer con dichas señales, dicho tratamiento en uno de los dominios podría resultar ser más simple que el otro, lo cual debe poder identificarse fácilmente, a fin de usar el más eficiente.

b. Fundamentos de la Transformada discreta de Fourier (DFT).-

b.1.- Definición.-

Para una señal continua $f(t)$ función del tiempo sobre el rango $-\infty < t < \infty$, la transformada de Fourier de la señal continua en el tiempo era definida por la ecuación 2.16, donde $-\infty < \omega < \infty$. La complicación matemática es que esta transformada requiere dominio de los elementos del cálculo diferencial e integral.

Para el caso de las señales de tiempo discreto y de duración finita, la Transformada aplicable se deriva simplemente de la transformada continua, reemplazando la integral por una suma finita.

De esta forma la Transformada Discreta de Fourier (algunas veces se la denota con el símbolo \mathcal{F}) es una transformación lineal reversible; es decir, se cumple la condición:

$$\mathcal{F} : \mathbb{C}^n \rightarrow \mathbb{C}^n \quad (2.21)$$

Donde \mathbb{C} denota números complejos.

En la relación 2.21 el elemento a transformar y el resultado de la transformación, son ambos complejos. En general pues, los números complejos $x(0), x(1) \dots, x(n-1)$ son transformados en los n números complejos $X(0), X(1) \dots, X(n-1)$ de acuerdo con la fórmula genérica:

$$X(\omega_k) \equiv \sum_{n=0}^{N-1} x(t_n) e^{-j\omega_k t_n} \quad (2.22)$$

$$\text{Con : } k = 0, 1, 2, \dots, N - 1.$$

Donde, debe notarse que:

\equiv Significa “es definida cómo” o “igual por definición”

$$\sum_{n=0}^{N-1} f(n) \equiv f(0) + f(1) + \dots + f(N-1)$$

$x(t_n) \equiv$ es la amplitud de la señal (real o compleja) en el tiempo t_n (seg)

$t_n \equiv nT = n$ -ésimo instante de muestreo (seg), donde n es entero ≥ 0

$T \equiv$ Intervalo de muestreo (seg.)

$X(\omega_k) \equiv$ es el espectro de x (valor complejo), a la frecuencia ω_k

$\omega_k \equiv k\Omega = k$ -ésima muestra de frecuencia (rad/seg)

$\Omega \equiv \frac{2\pi}{NT} =$ Intervalo de muestreo de frecuencia en radianes (rad/seg)

$f_s \equiv \frac{1}{T} =$ Frecuencia de muestreo (muestras/seg, o Hertz - Hz)

$N \equiv$ Número de muestras (entero)

El intervalo de muestreo T es también llamado período de muestreo.

Cuando todas las N muestras de la señal $x(t_n)$ son reales, se puede decir que $\mathbf{x} \in \mathbf{R}^N$

Por otro lado, si x fuera un número complejo, $\mathbf{x} \in \mathbf{C}^N$

b.2.- Demostración.-

Por lo tanto, la serie de Fourier discreta es simplemente una modificación de la serie de Fourier tradicional, pero sustituyendo las integrales por sumatorias de las muestras, y el periodo ahora en vez de ser T (número real) será N , siendo N un número entero (lo que le da el carácter discreto), de forma que se define la serie de Fourier discreta como:

$$x[n] = \sum_{n=-\infty}^{n=\infty} a_k e^{jk\Omega t} \quad (2.23)$$

Donde:

$$\Omega = \frac{2\pi}{N}$$

Para la cual, se cumplirá ahora que $x[n]=x[n+N]$, puesto que N es el periodo fundamental.

Para obtener los coeficientes del desarrollo en serie de Fourier haremos:

$$a_k = \frac{1}{N} \sum_{n=-\infty}^{n=\infty} x[n] e^{jk\Omega n} \quad (2.24)$$

Tal como pasa en el caso continuo, la serie de Fourier discreta es aplicable solamente a señales periódicas. Para señales aperiódicas aplicamos la transformada de Fourier discreta, que podemos deducir a partir de la transformada de Fourier continua.

Sea el siguiente sistema mostrado en la Fig. 2.7, que muestrea la señal $x(t)$ con la señal de pulsos $p(t)$:

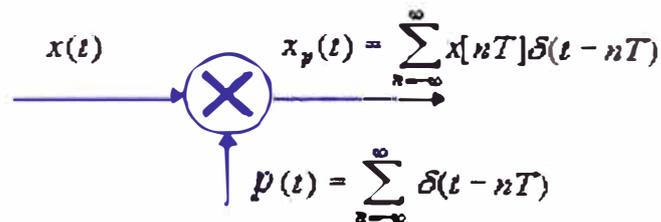


Fig. 2.7.- Sistema Muestreo de Señal $x(t)$

$$\begin{aligned}
 x_p(\omega) &= \int_{-\infty}^{\infty} x_p(t) e^{-j\omega t} dt = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[nT] \delta(t - nT) e^{-j\omega t} dt \\
 x_p(\omega) &= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[nT] e^{-j\omega t} \delta(t - nT) dt = \\
 x_p(\omega) &= \sum_{n=-\infty}^{\infty} x[nT] e^{-j\omega nT} \int_{-\infty}^{\infty} \delta(t - nT) dt \quad (2.25) \\
 x_p(\omega) &= \sum_{n=-\infty}^{\infty} x[nT] e^{-j\omega nT}
 \end{aligned}$$

$$X_p(\Omega) = \sum_{n=-\infty}^{n=\infty} x[n] e^{j\Omega n} \quad (2.26)$$

Pasamos de tiempo continuo a tiempo discreto con un cambio de variable $\{\Omega \Rightarrow \omega T\}$ (la que ahora resulta ser una variable discreta), y obtenemos la transformada discreta de Fourier (DFT) de $x[n]$ (7, 8, 9).

Es necesario tener en consideración el hecho que, para que la señal pueda ser procesada por sistemas reales, es necesario limitar la duración de la señal $x[n]$, ya que no puede extenderse al infinito. Sin embargo, las condiciones a establecer deben ser tales que garanticen un recupero preciso de la señal.

La ventaja fundamental de esta fórmula es que no requiere de los elementos del cálculo diferencial e integral para definir el DFT (incluso para su inversa), y solo se trata de sumatorias finitas. Para el procesamiento de señal digital, donde las señales y el espectro son procesados únicamente en forma muestreada, el DFT es lo adecuado (y como veremos más adelante, una forma más eficiente dada por la Transformada Rápida de Fourier – FFT), por ser más simple matemáticamente; y además, más relevante computacionalmente que la Transformada de Fourier continua.

b.3.- Matemáticas del DFT.-

En los tratados de procesamiento digital de señales, es común escribir la DFT y su inversa en la forma que se describe a continuación, fórmulas obtenidas haciendo $T=1$ en la definición previa.

(2.27)

$$X(\omega_k) \equiv \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N}, \text{ donde } : k = 0, 1, 2, \dots, N-1. \quad (2.28)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi nk/N}, \text{ donde } : n = 0, 1, 2, \dots, N-1.$$

Donde $x(n)$ denota la señal de entrada en el tiempo (instante o muestra) n , y $X(k)$ denota la muestra espectral k -ésima.

Hay dos elementos en la notación del DFT que es necesario definir:

$$\begin{aligned}
 j &\triangleq \sqrt{-1} \\
 e &\triangleq \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = 2.71828182845905\dots \\
 j &= \sqrt{-1}
 \end{aligned}
 \tag{2.29}$$

El primero, $j = \sqrt{-1}$, es la base para los números complejos.

El segundo, $e = 2.718\dots$, es un número trascendental real definido por el límite indicado.

En la formulación vemos la complejidad adicional, de que los números complejos aparecen en los exponentes, como en la expresión:

$$s_k(n) \triangleq e^{j2\pi nk/N} \tag{2.30}$$

Es posible desarrollar sistemáticamente lo que definimos como exponente imaginario con el fin de definir tal expresión matemática.

Matemáticamente, con e^j , para entender los exponentes imaginarios, podemos ir a la identidad de Euler:

$$e^{j\theta} = \cos(\theta) + j \sin(\theta) \tag{2.31}$$

Esta identidad es la clave para entender el significado de las expresiones como:

$$s_k(t_n) \triangleq e^{j\omega_k t_n} = \cos(\omega_k t_n) + j \sin(\omega_k t_n). \tag{2.32}$$

Se puede apreciar que tal expresión define una muestra de una sinusoidal compleja, y esta representación es importante, dado que representa muchas señales en la realidad.

Finalmente, es necesario entender que la sumatoria sobre n define la DFT. Se entiende que el deberá ser visto como el cómputo de un producto interno de la señal x y S_k definido arriba, de forma que podamos escribir el DFT, usando la notación de productos internos, como:

$$X(k) \triangleq \langle x, s_k \rangle \quad (2.33)$$

$$s_k(n) \triangleq e^{j2\pi nk/N}$$

Donde $s_k(n)$ es la muestra compleja del senoide en frecuencia en radianes (normalizado) $\omega_k = 2\pi k/N$, y la operación del producto interno $\langle \cdot, \cdot \rangle$ es definida por:

$$\langle x, y \rangle \triangleq \sum_{n=0}^{N-1} x(n) \overline{y(n)}. \quad (2.34)$$

Demostremos que el producto interno de x con k -ésimo de la senoide base s_k es una medida de cuanto de s_k está presente en x y en que fase (esto porque es un número complejo).

Después de seguir adelante, se considera el hecho de que la DFT inversa puede ser entendida como la suma de las proyecciones de x sobre $\{s_k\}_{k=0}^{N-1}$, es decir, se demuestra que:

$$x(n) = \sum_{k=0}^{N-1} \tilde{X}_k s_k(n), \quad n = 0, 1, 2, \dots, N-1 \quad (2.35)$$

Donde:

$$\tilde{X}_k \triangleq \frac{X(k)}{N}$$

es el coeficiente de proyección de x sobre

Introduciendo la notación $x \triangleq x(\cdot)$ para referirse a la señal completa $x(n)$ para todo $n \in [0, N - 1]$, la IDFT puede ser escrita más simplemente como:

$$x = \sum_k \tilde{X}_k s_k. \quad (2.36)$$

Notar que ambos sinusoides básicas s_k y su coeficiente de proyección \tilde{X}_k son valores complejos en general.

Teniendo un entendimiento completo del DFT y su matemáticamente inversa, podemos probar varios Teoremas de Fourier, así como el “teorema del desplazamiento”, el “teorema de convolución” y el “teorema de Parseval”.

Los teoremas de Fourier proveen los términos básicos para trabajar con señales en los dominios del tiempo y de la frecuencia. Estos pueden ser usados para responder cuestiones tales como: ¿Que pasa en el dominio de la frecuencia si efectúo una operación en x en el dominio del tiempo?

Finalmente, estudiamos una variedad de ejemplos prácticos de análisis de espectros, usando primariamente el lenguaje de programación MatLAB para analizar y diagramar señales y sus espectros.

2.3.5. Transformada discreta de Fourier inversa (IDFT)

La transformada discreta de Fourier inversa (IDFT) está dada por:

$$x(t_n) = \frac{1}{N} \sum_{k=0}^{N-1} X(\omega_k) e^{j\omega_k t_n}, \quad n = 0, 1, 2, \dots, N - 1. \quad (2.37)$$

[La DFT inversa es escrita usando el símbolo '—' en vez de 'Δ', porque el resultado sigue de la definición de la DFT.]

2.3.6. Transformada Rápida de Fourier (FFT)

a. Conceptos.-

En la sección previa se plasmaron los conceptos de la transformada Discreta de Fourier (DFT), la cual es usada para producir análisis de frecuencia de señales no periódicas no discretas. Sin embargo, y desde el punto de vista computacional, si se observa la ecuación para la DFT, se puede apreciar que es aún bastante complicada para ser ejecutada, dado que involucra muchas sumas y multiplicaciones de números complejos. Así, una simple señal de 8 muestras requeriría para ejecutar el DFT de 49 multiplicaciones complejas y 56 sumas complejas. Si bien a este nivel es todavía manejable, para una señal real que podría tener 1024 muestras, la cual requeriría más de 20 millones de multiplicaciones y sumas complejas, representaría una gran capacidad de procesamiento y tiempo para realizar el mismo, que podría resultar inmanejable.

Ante esto, se presenta la metodología de la Transformada Rápida de Fourier (FFT) como una alternativa metódica para alcanzar el mismo resultado, pero con menos sobrecarga en los cálculos involucrados. Así pues, la FFT es simplemente un algoritmo eficiente para calcular la transformada discreta de Fourier (DFT) y su transformada inversa. Es por tanto de gran importancia para una amplia variedad de aplicaciones, desde el procesamiento digital de señales (filtrado lineal, análisis de correlación, análisis de espectros, etc.) hasta la resolución de ecuaciones diferenciales parciales; así como, algoritmos para la rápida multiplicación de enteros largos. Como se mencionó antes, para que la señal pueda ser procesada por sistemas reales, es necesario limitar la duración de la señal $x[n]$, ya que no puede extenderse al infinito. Sin embargo, las condiciones a establecer deben ser tales que garanticen una recuperación lo más precisa posible de la señal original (2, 3, 4, 10,11).

b. Matemáticas en el FFT.-

Retornando al problema computacional de la DFT, ésta consiste en computar la secuencia $\{X(k)\}$ de N números complejos dando otra secuencia de datos $\{x(n)\}$ de longitud N , de acuerdo a la fórmula:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad (2.38)$$

Donde : $0 \leq k \leq N - 1$

$$W_N = e^{-j2\pi/N}$$

En general, los datos de la secuencia $x(n)$ también se asume que serían compuestas por valores complejos.

Similarmente, la DFT inversa (IDFT) llega a ser:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-nk} \quad (2.39)$$

Donde : $0 \leq n \leq N - 1$

Siendo que tanto la DFT y la IDFT involucra básicamente el mismo tipo de computaciones, el tratamiento sobre los algoritmos de computación eficiente para el DFT también es aplicable a la computación eficiente del IDFT.

Se observa que para cada valor de k , la computación directa de $X(k)$ involucra N multiplicaciones complejas ($4N$ multiplicaciones reales) y $N-1$ sumas complejas ($4N-2$ sumas reales). Consecuentemente, para computar todos los N valores de la DFT se requieren N^2 multiplicaciones complejas y N^2-N sumas complejas.

La computación directa de la DFT es básicamente ineficiente primariamente porque ésta no explota las propiedades de simetría y periodicidad del factor fase de W_N . En particular, estas propiedades son:

- Propiedad de la simetría:

$$W_N^{k+N/2} = -W_N^k$$

- Propiedad de la periodicidad:

$$W_N^{k+N} = W_N^k$$

Los algoritmos de computación eficiente descrita en ésta sección, conocida colectivamente como Algoritmos de Transformada Rápida de Fourier (FFT), explotan precisamente esas dos propiedades básicas del factor de fase.

b. Algoritmos FFT.-

b.1.- Radix-2.-

Consideremos la computación del DFT desde el punto de vista “*divide y conquista*”, para lo cual se tiene: $N=2^v$. Por tanto dividimos la secuencia de datos de N puntos en dos secuencias de datos de $N/2$ puntos, a las que denominaremos $f_1(n)$ y $f_2(n)$, correspondiendo a las muestras numeradas pares e impares respectivamente, es decir:

$$\begin{aligned} f_1(n) &= x(2n) \\ f_2(n) &= x(2n+1), n = 0, 1, \dots, N/2 - 1 \end{aligned} \quad (2.40)$$

Así, $f_1(n)$ y $f_2(n)$ son obtenidas por la decimación de $x(n)$ por un factor de 2, y por tanto, el algoritmo FFT resultante es llamado un algoritmo de decimación en tiempo.

Ahora el DFT de N puntos puede ser expresado en términos de las DFT's de las secuencias decimadas como sigue:

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{nk}, k = 0, 1, \dots, N-1 \\ X(k) &= \sum_{n=even} x(n)W_N^{nk} + \sum_{n=odd} x(n)W_N^{nk} \\ X(k) &= \sum_{m=0}^{(N/2)-1} x(2m)W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1)W_N^{k(2m+1)} \end{aligned} \quad (2.41)$$

Pero $W_N^2 = W_{N/2}$. Con ésta sustitución, la ecuación puede ser expresada como:

$$\begin{aligned} X(k) &= \sum_{m=0}^{(N/2)-1} f_1(m)W_{N/2}^{mk} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m)W_{N/2}^{km} \\ X(k) &= F_1(k) + W_N^k F_2(k), k = 0, 1, \dots, N-1 \end{aligned} \quad (2.42)$$

Donde $F_1(k)$ y $F_2(k)$ son las DFT's de $N/2$ puntos de las secuencias $f_1(m)$ y $f_2(m)$, respectivamente.

Dado que $F_1(k)$ y $F_2(k)$ son periódicas, con periodo $N/2$, se tiene $F_1(k+N/2) = F_1(k)$ y $F_2(k+N/2) = F_2(k)$. Adicionalmente, se tiene el factor $W_N^{k+N/2} = -W_N^k$.

Entonces, la ecuación puede ser expresada como:

$$\begin{aligned} X(k) &= F_1(k) + W_N^k F_2(k), k = 0, 1, \dots, N/2 - 1 \\ X(k + N/2) &= F_1(k) - W_N^k F_2(k), k = 0, 1, \dots, N/2 - 1 \end{aligned} \quad (2.44)$$

Se observa que el cómputo directo de $F_1(k)$ requiere de $(N/2)^2$ multiplicaciones complejas. Lo mismo se aplica a $F_2(k)$. Así mismo, existen $N/2$ multiplicaciones complejas adicionales para computar $W_N^k F_2(k)$. Entonces el cómputo de $X(k)$ requiere $2(N/2)^2 + N/2 = N^2/2 + N/2$ multiplicaciones complejas. El primer paso resulta en una reducción del número de multiplicaciones desde N^2 a $N^2/2 + N/2$, el cuál es cerca de un factor de 2 por N de longitud.

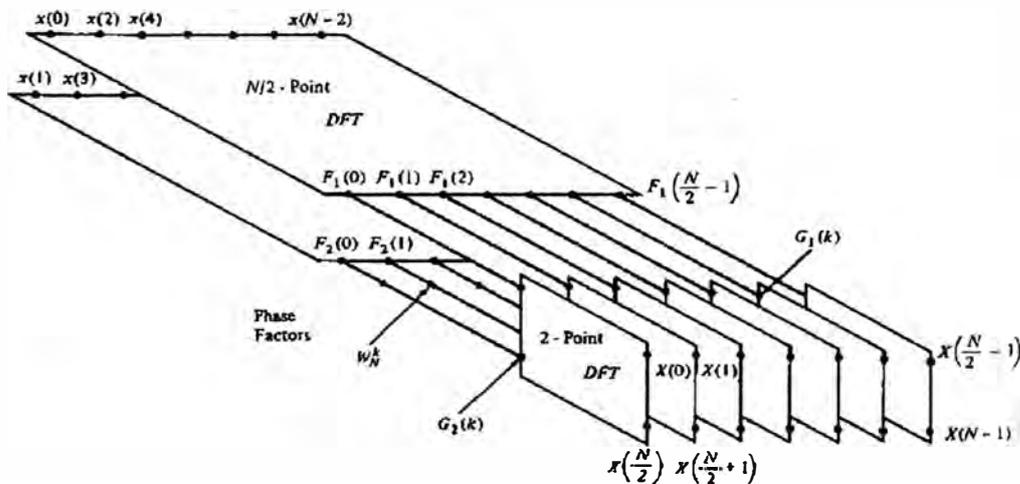


Fig. 2.8.- El primer paso en el algoritmo de la decimación en tiempo

Computando las DFT's de $N/4$ puntos, se puede obtener las DFT's de $N/2$ puntos de $F_1(k)$ y $F_2(k)$ desde las relaciones:

$$\begin{aligned}
 F_1(k) &= F\{f_1(2n)\} + W_{N/2}^k F\{f_1(2n+1)\}, k = 0, 1, \dots, N/4 - 1, n = 0, 1, \dots, N/4 - 1 \\
 F_1\left(k + \frac{N}{4}\right) &= F\{f_1(2n)\} - W_{N/2}^k F\{f_1(2n+1)\}, k = 0, 1, \dots, N/4 - 1, n = 0, 1, \dots, N/4 - 1 \\
 F_2(k) &= F\{f_2(2n)\} + W_{N/2}^k F\{f_2(2n+1)\}, k = 0, 1, \dots, N/4 - 1, n = 0, 1, \dots, N/4 - 1 \\
 F_2\left(k + \frac{N}{4}\right) &= F\{f_2(2n)\} - W_{N/2}^k F\{f_2(2n+1)\}, k = 0, 1, \dots, N/4 - 1, n = 0, 1, \dots, N/4 - 1
 \end{aligned}
 \tag{2.45}$$

Donde: $F\{*\}$ representa la transformada de Fourier

La decimación de la secuencia de datos puede ser repetida muchas veces hasta que resulte en secuencias reducidas a secuencias de un punto. Para $N = 2^v$, esta decimación puede ser ejecutada $v = \log_2 N$ veces. Así, el número total de multiplicaciones complejas es reducida a $(N/2)\log_2 N$. El número de sumas complejas es: $N\log_2 N$.

Para propósitos ilustrativos, la Fig. 2.9 muestra el cómputo de una DFT de $N=8$ puntos. Se observa que el cómputo es realizado en tres etapas, comenzando con el cómputo de 4 DFT's de 2 puntos, luego 2 DFT's de 4 puntos, y finalmente, una DFT de 8 puntos. La combinación de los DFT's más pequeños para formar el DFT mayor es ilustrada en la Fig. 2.9 para $N=8$.

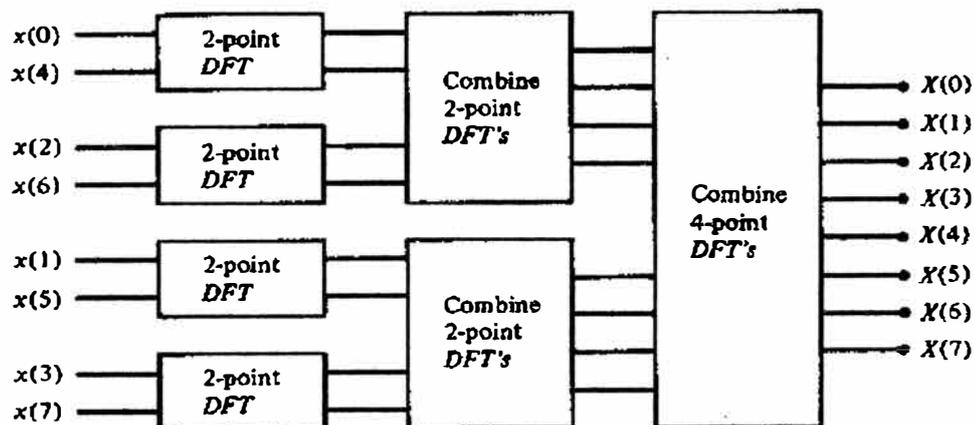


Fig. 2.9.- Tres etapas en la computación de DFT $N = 8$ -puntos

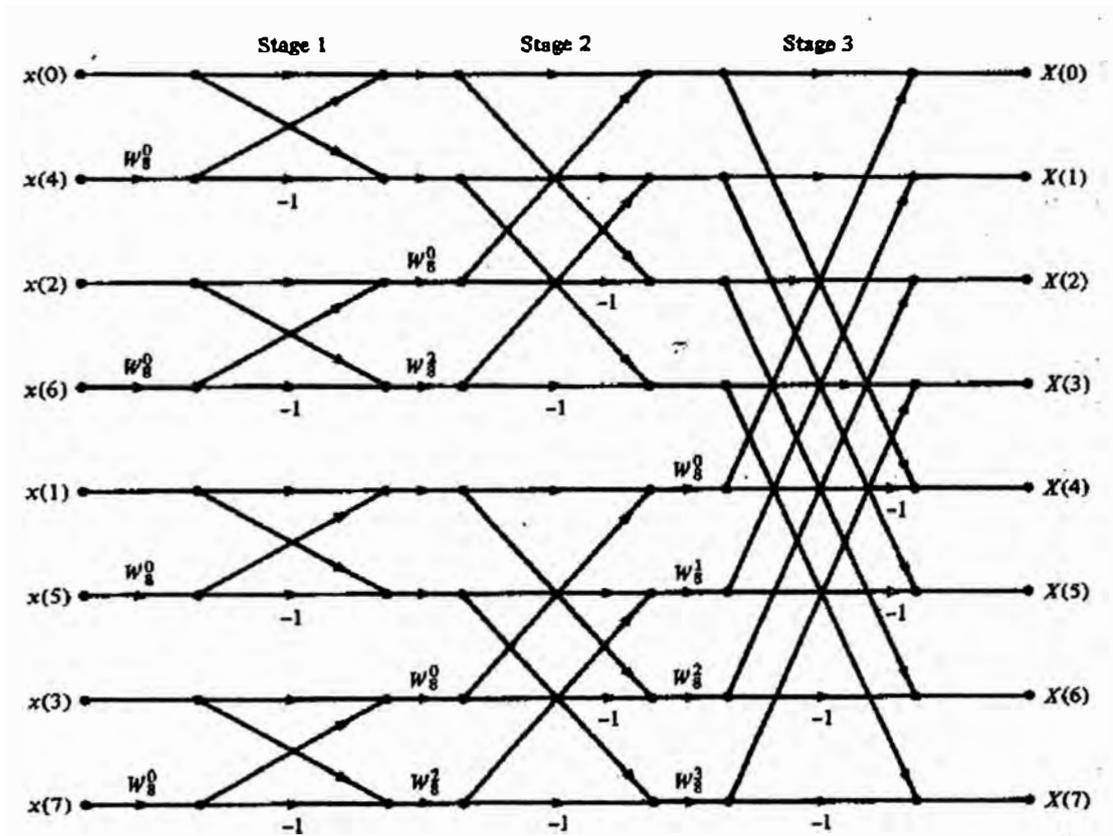


Fig. 2.10.- Algoritmo FFT - Decimación en tiempo de 8 puntos

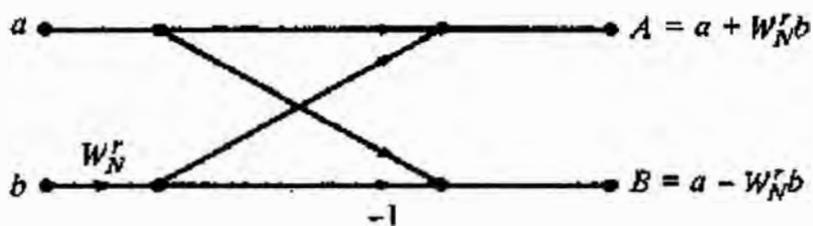
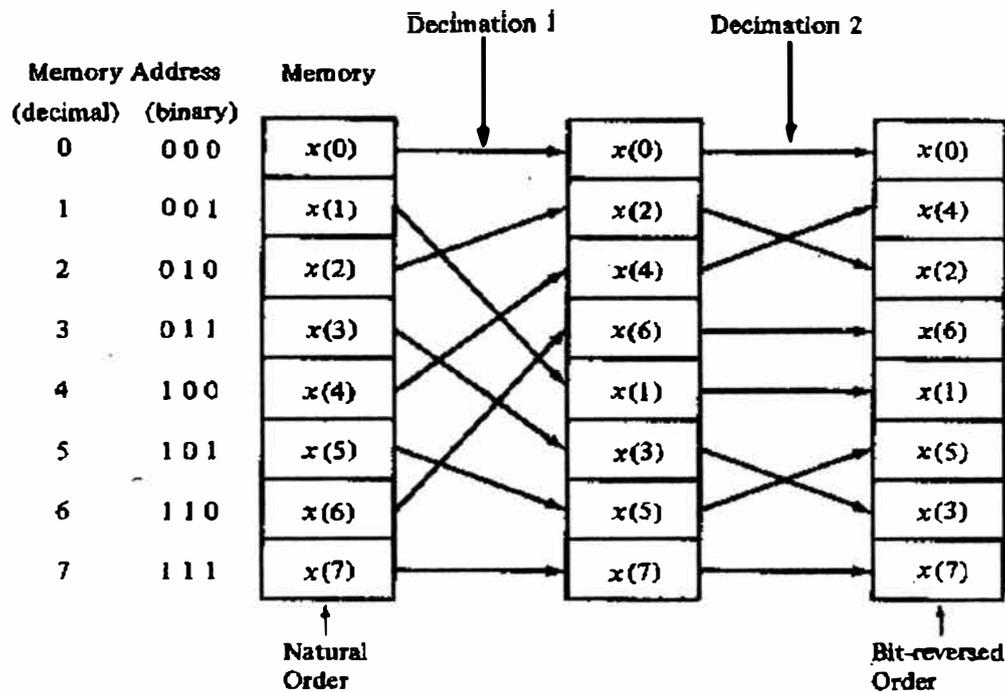


Fig. 2.11.- Algoritmo FFT - en decimación en tiempo Computación Básica Mariposa (Butterfly)

Una observación importante es referida al orden de la secuencia de datos de entrada después de decimación en $(v-1)$ veces. Por ejemplo, si se considera el caso donde $N=8$, se sabe que la primera decimación conlleva las secuencias $x(0), x(2), x(4), x(6), x(1), x(3), x(5), x(7)$, y la segunda decimación resulta en la secuencia $x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)$.

Este barajar la secuencia de datos de entrada tiene un orden bien definido como se puede verificar observando la Fig. 2.12, la cual ilustra la decimación de una secuencia de 8 puntos.



$$\begin{array}{l}
 \overline{(n_2 n_1 n_0)} \rightarrow (n_0 n_2 n_1) \rightarrow (n_0 n_1 n_2) \\
 \hline
 (000) \rightarrow (000) \rightarrow (000) \\
 (001) \rightarrow (100) \rightarrow (100) \\
 (010) \rightarrow (001) \rightarrow (010) \\
 (011) \rightarrow (101) \rightarrow (110) \\
 (100) \rightarrow (010) \rightarrow (001) \\
 (101) \rightarrow (110) \rightarrow (101) \\
 (110) \rightarrow (011) \rightarrow (011) \\
 (111) \rightarrow (111) \rightarrow (111)
 \end{array}$$

(b)

Fig. 2.12.- Barajado de datos y bit de inversa

Otro algoritmo FFT Radix-2 importante, llamado algoritmo de decimación de frecuencia, es obtenido también usando la táctica de "divide y conquista". Para derivar el algoritmo, se inicia por separar la fórmula del DFT en dos sumatorias, una que incluye la suma sobre los primeros datos de los $N/2$ puntos y la segunda suma incluyendo los últimos $N/2$ puntos. Así se obtiene:

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n)W_N^{nk} + \sum_{n=0}^{(N/2)-1} x(n)W_N^{nk} \quad (2.46)$$

$$X(k) = \sum_{m=0}^{(N/2)-1} x(m)W_N^{km} + W_N^{Nk/2} \sum_{m=0}^{(N/2)-1} x\left(m + \frac{N}{2}\right)W_N^{km}$$

Dado que :

$$W_N^{kN/2} = (-1)^k$$

$$X(k) = \sum_{m=0}^{(N/2)-1} \left[x(m) + (-1)^k x\left(m + \frac{N}{2}\right) \right] W_N^{km}$$

Si se separa (decima) $X(k)$ en las muestras pares e impares. Entonces obtenemos:

$$X(2k) = \sum_{m=0}^{(N/2)-1} \left[x(m) + x\left(m + \frac{N}{2}\right) \right], k = 0, 1, \dots, N/2 - 1 \quad (2.47)$$

$$X(2k+1) = \sum_{m=0}^{(N/2)-1} \left[x(m) - x\left(m + \frac{N}{2}\right) \right], k = 0, 1, \dots, N/2 - 1$$

Donde se usa el hecho que $W_N^2 = W_{N/2}$.

El procedimiento computacional arriba descrito puede ser repetido a través de la decimación de las DFT's de $N/2$ puntos $X(2k)$ y $X(2k+1)$. El proceso entero incluye $v = \log_2 N$ etapas de decimación, donde cada etapa incluye $N/2$ mariposas (butterflies) del tipo mostrado en la Fig. 3.14.

Consecuentemente, el cómputo del DFT de N puntos vía la FFT con decimación de frecuencia requiere $(N/2)\log_2 N$ multiplicaciones complejas y $N\log_2 N$ sumas complejas, tal como la decimación en tiempo. Con propósitos ilustrativos, el algoritmo de decimación en frecuencia de 8 puntos es muestra en las figuras 2.13 y 2.14.

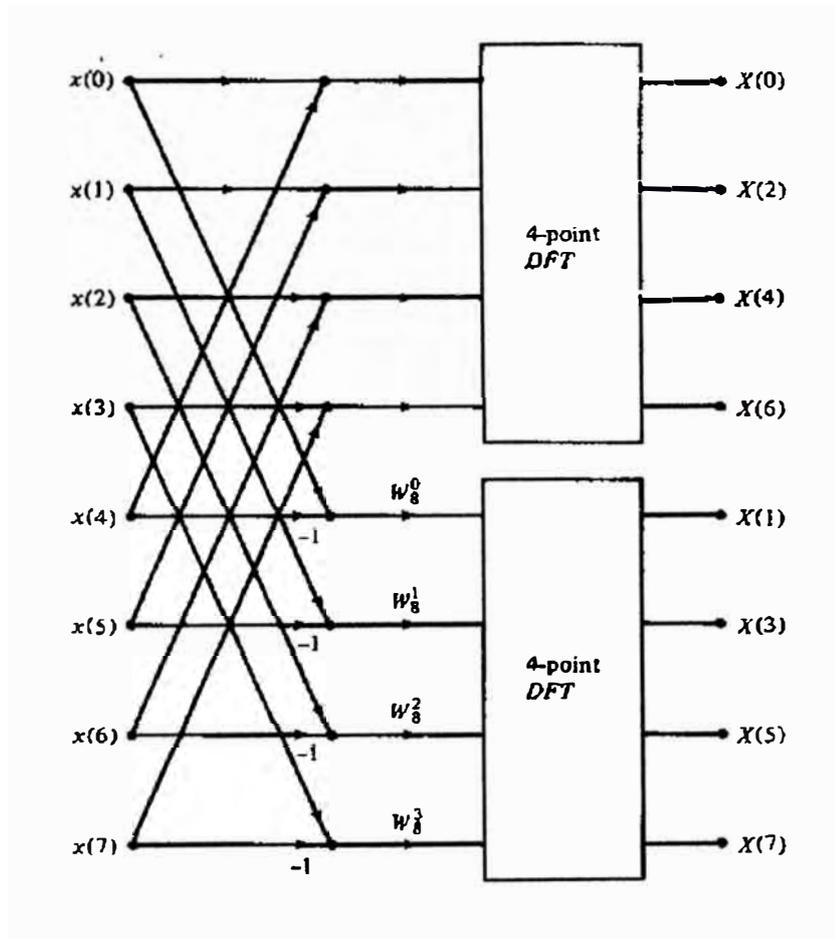


Fig. 2.13.- Algoritmo FFT – Decimación en frecuencia primera etapa

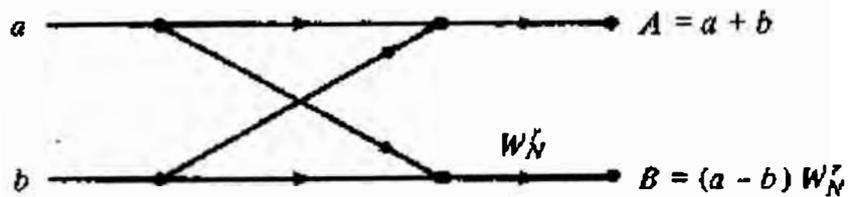


Fig. 2.14.- Computación Mariposa (butterfly) - Decimación en frecuencia

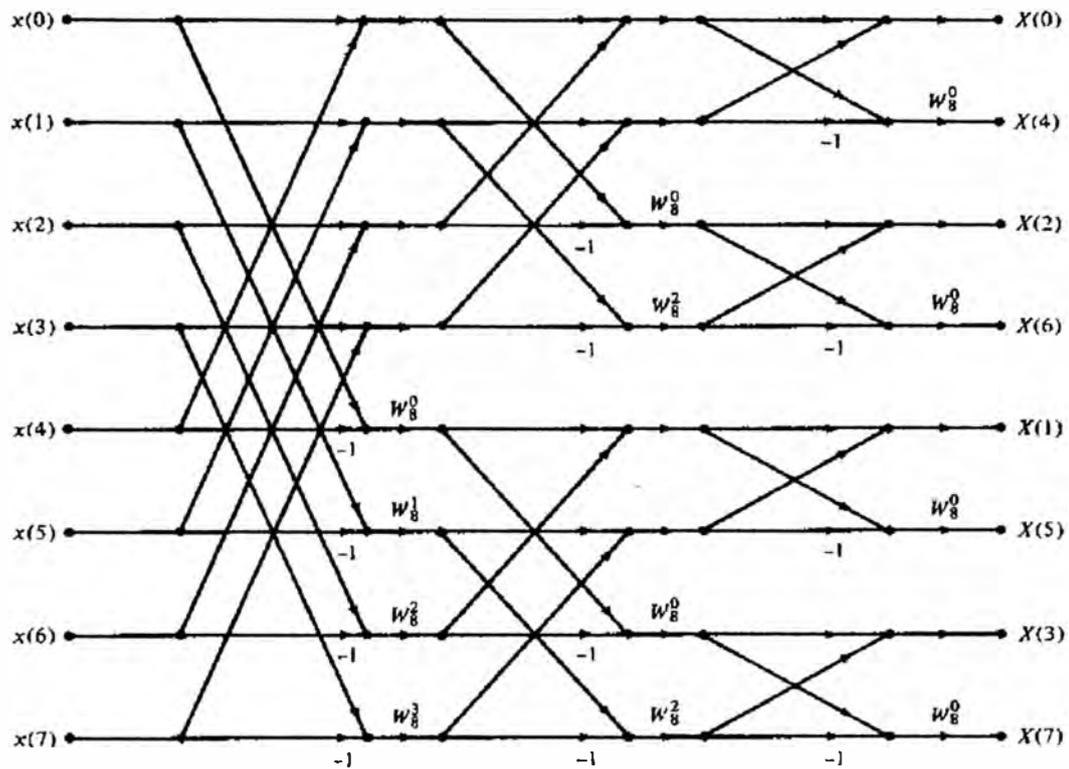


Fig. 2.15.- Algoritmo FFT - Decimación en frecuencia $N=8$ puntos

Observamos de la figura 2.15 que los datos de entrada $x(n)$ ocurren en un orden natural, pero la salida DFT ocurre en orden inverso. Se nota también que los cálculos son ejecutados en el lugar. Sin embargo, es posible reconfigurar el algoritmo de decimación en frecuencia de forma que la secuencia de entrada ocurra en un orden de *bit inverso* mientras que el DFT de la salida ocurre en un orden normal. Adicionalmente, si se deja de lado el requerimiento que los cálculos hechos en su lugar, es también posible tener ambas, los DFT de las entradas y las salidas, en el orden normal.

b.2.- FFT Radix-4.-

Cuando el número puntos de datos N en el DFT es una potencia de 4 (es decir, $N = 4^v$), se puede, por supuesto usar siempre el algoritmo radix-2 para el cómputo. Sin embargo, para este caso, es mucho más eficiente computacionalmente emplear el algoritmo radix- r FFT.

Se inicia describiendo brevemente un algoritmo radix-4 con decimación en tiempo. Para esto, se separa o decima la secuencia de entrada de N puntos en unas 4 secuencias: $x(4n)$, $x(4n+1)$, $x(4n+2)$, $x(4n+3)$, $n = 0, 1, \dots, N/4-1$.

$$X(p, q) = \sum_{l=0}^3 [W_N^{lq} F(l, q)] W_4^{lp} \quad (2.48)$$

$$F(l, q) = \sum_{m=0}^{(N/4)-1} x(l, m) W_{N/4}^{lq}$$

$$p = 0, 1, 2, 3; l = 0, 1, 2, 3; q = 0, 1, 2, \dots, N/4 - 1$$

y

$$x(l, m) = x(4m + 1)$$

$$X(p, q) = X\left(\frac{N}{4} p + q\right)$$

Así las 4 DFT's de $N/4$ puntos $F(l, q)$ obtenida de la ecuación arriba descrita es combinada para portar la DFT de N puntos. La expresión de combinar las DFT's de $N/4$ puntos define la decimación en tiempo mariposa radix-4, la cual puede ser expresada en forma matricial tal como sigue:

$$\begin{bmatrix} X(0, q) \\ X(1, q) \\ X(2, q) \\ X(3, q) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} W_N^0 F(0, q) \\ W_N^q F(1, q) \\ W_N^{2q} F(2, q) \\ W_N^{3q} F(3, q) \end{bmatrix} \quad (2.49)$$

La mariposa radix-4 es graficada en la Figura 3.16a y en una forma más compacta en la Fig. 3.16b. Notar que cada mariposa incluye tres multiplicaciones complejas, desde que $W_N^0 = 1$ y 12 sumas complejas.

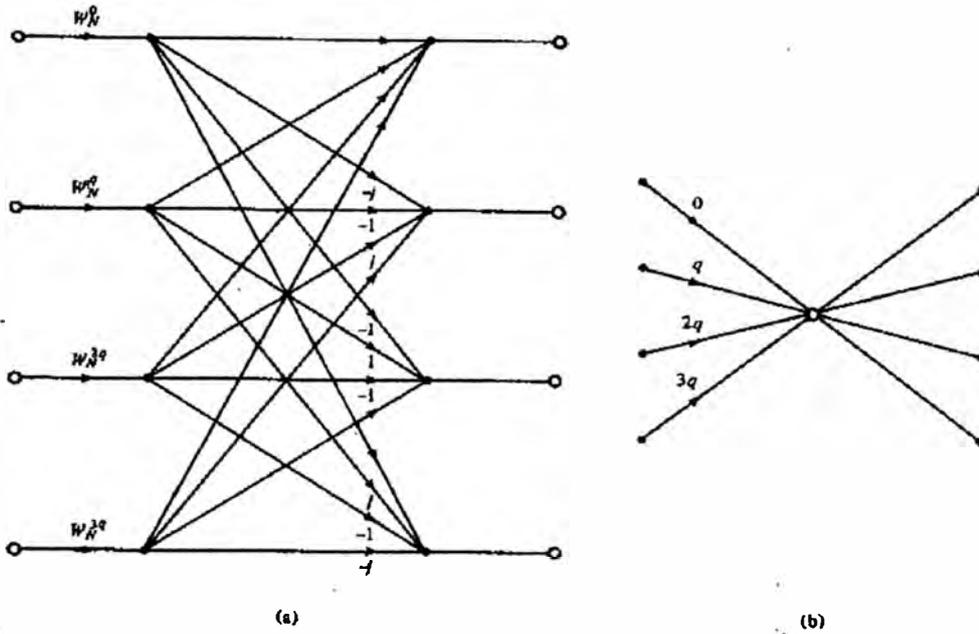


Fig. 2.16.- Algoritmo FFT – Computación Mariposa (Butterfly) Radix-4

Ejecutando las sumas en dos pasos, es posible reducir el número de sumas por mariposa desde 12 a 8. Esto puede ser realizado expresando la matriz de la transformación lineal mencionada previamente como un producto de dos matrices como sigue:

$$\begin{bmatrix} X(0,q) \\ X(1,q) \\ X(2,q) \\ X(3,q) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} W_N^0 F(0,q) \\ W_N^1 F(1,q) \\ W_N^2 F(2,q) \\ W_N^3 F(3,q) \end{bmatrix} \quad (2.50)$$

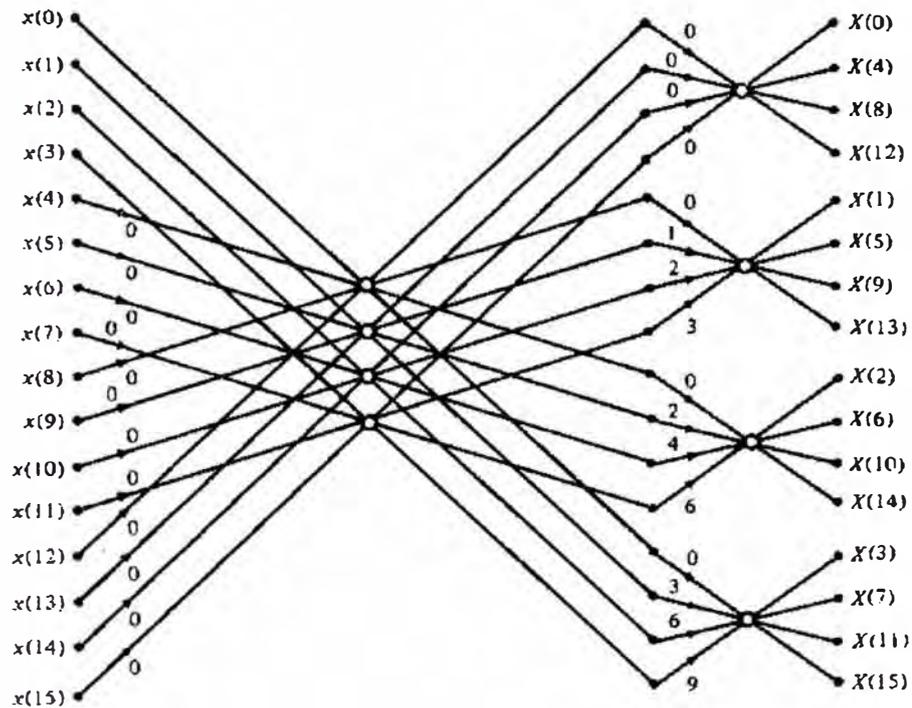


Fig. 2.17.- Algoritmo FFT con decimación en tiempo Radix-4 con entrada en orden normal y salida en digito orden invertido

Un Radix-4 con decimación en frecuencia de 16 puntos se muestra en la figura 2.18. Su entrada es en orden normal y su salida es en orden de digito invertido. Este tiene exactamente la misma complejidad computacional de la decimación en tiempo del algoritmo radix-4 FFT.

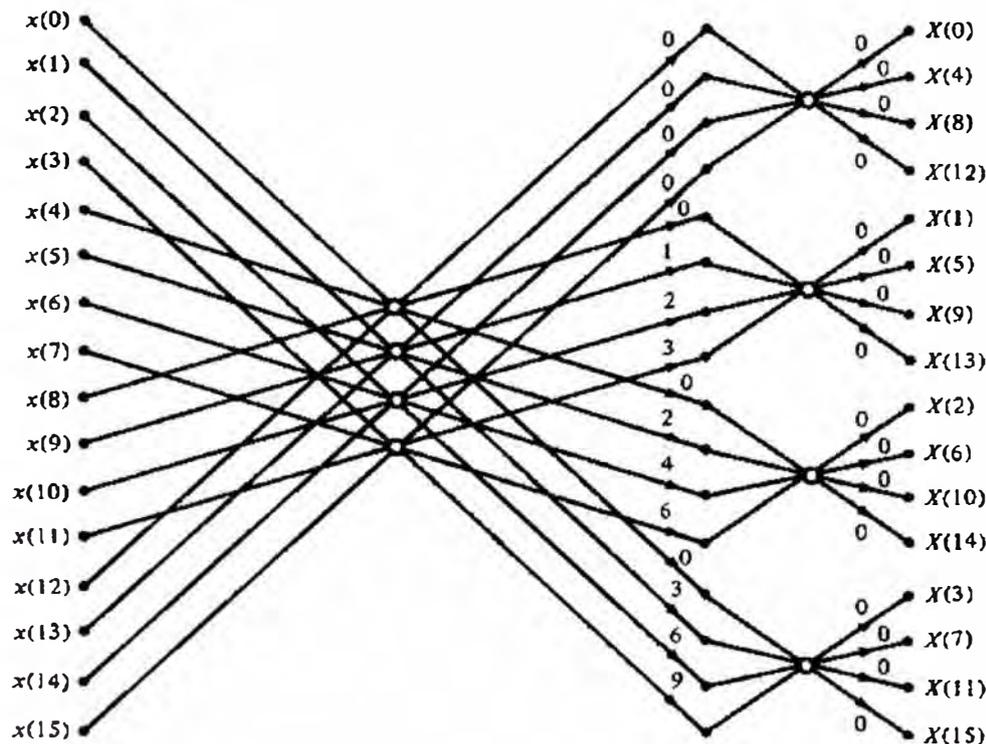


Figura 2.18.- Algoritmo FFT con decimación en tiempo Radix-4 con entrada en orden normal y salida en digito orden invertido para 16 puntos

Para propósitos ilustrativos, usamos la rederivación el algoritmo con decimación en frecuencia radix-4 rompiendo la formula del DFT para N puntos en 4 DFT's más pequeños. Se tiene entonces:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}$$

$$X(k) = \sum_{n=0}^{N/4-1} x(n)W_N^{kn} + \sum_{n=N/4}^{N/2-1} x(n)W_N^{kn} + \sum_{n=N/2}^{3N/4-1} x(n)W_N^{kn} + \sum_{n=3N/4}^{N-1} x(n)W_N^{kn} \quad (2.51)$$

$$X(k) = \sum_{n=0}^{N/4-1} x(n)W_N^{kn} + W_N^{Nk/4} \sum_{n=0}^{N/4-1} x\left(n + \frac{N}{4}\right)W_N^{kn}$$

$$\Rightarrow \Rightarrow +W_N^{Nk/2} \sum_{n=0}^{N/4-1} x\left(n + \frac{N}{2}\right)W_N^{kn} + W_N^{3Nk/4} \sum_{n=0}^{N/4-1} x\left(n + \frac{3N}{4}\right)W_N^{kn}$$

De la definición de los factores *twiddle*, se tiene:

$$W_N^{kN/4} = (-j)^k, W_N^{kN/2} = (-1)^k, W_N^{3kN/4} = (j)^k$$

Así :

$$X(k) = \sum_{n=0}^{N/4-1} \left[x(n) + (-j)^k x\left(n + \frac{N}{4}\right) + (-j)^k x\left(n + \frac{N}{2}\right) + (-j)^k x\left(n + \frac{3N}{4}\right) \right] W_N^{nk} \quad (2.52)$$

La relación no es un DFT de $N/4$ puntos porque el factor twiddle depende de N y no de $N/4$. Para convertirlo en una DFT de $N/4$ puntos se subdivide la secuencia DFT en 4 secuencias $N/4$ subsecuencias, $X(4k)$, $X(4k+1)$, $X(4k+2)$, y $X(4k+3)$, $k = 0, 1, \dots, N/4$. Así se obtiene el DFT radix-4 con decimación en frecuencia como:

$$\begin{aligned} X(4k) &= \sum_{n=0}^{N/4-1} \left[x(n) + x\left(n + \frac{N}{4}\right) + x\left(n + \frac{N}{2}\right) + x\left(n + \frac{3N}{4}\right) \right] W_N^0 W_{N/4}^{kn} \\ X(4k+1) &= \sum_{n=0}^{N/4-1} \left[x(n) - jx\left(n + \frac{N}{4}\right) - x\left(n + \frac{N}{2}\right) + jx\left(n + \frac{3N}{4}\right) \right] W_N^n W_{N/4}^{kn} \\ X(4k+2) &= \sum_{n=0}^{N/4-1} \left[x(n) - x\left(n + \frac{N}{4}\right) + x\left(n + \frac{N}{2}\right) - x\left(n + \frac{3N}{4}\right) \right] W_N^{2n} W_{N/4}^{kn} \\ X(4k+3) &= \sum_{n=0}^{N/4-1} \left[x(n) + jx\left(n + \frac{N}{4}\right) - x\left(n + \frac{N}{2}\right) - jx\left(n + \frac{3N}{4}\right) \right] W_N^{3n} W_{N/4}^{kn} \end{aligned} \quad (2.53)$$

Donde se ha usado la propiedad de $W_N^{4kn} = W_{N/4}^{kn}$.

Notar que la entrada de cada DFT de $N/4$ puntos es una combinación lineal de 4 muestras de señal escaladas por el factor *twiddle*. Este procedimiento es repetido v veces, donde $v = \log_4 N$.

b.3.- FFT Split Radix.-

Una inspección del flujo-grama del radix-2 con decimación en frecuencia mostrado en la figura 2.15 indica que los puntos pares del DFT pueden ser calculados independientemente de los puntos impares.

Esto sugiere entonces la posibilidad de usar métodos computacionales diferentes para las partes independientes del algoritmo, y esto con el objetivo de reducir el número de cálculos.

El algoritmo FFT radix-split (SRFFT) explota la idea de usar en combinación tanto el radix-2 y el radix-4 en el mismo algoritmo FFT.

Entonces, si recordamos en el algoritmo FFT con decimación de frecuencia del radix-2, las muestras pares del DFT de N puntos está dada por:

$$X(2k) = \sum_{n=0}^{N/2-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] W_{N/2}^n, \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (2.54)$$

Un radix-2 es suficiente para este cómputo.

Las muestras impares $\{X(2k+1)\}$ del DFT requieren una pre-multiplicación de la secuencia de entrada con los factores *twiddle* W_N^n .

Para estas muestras, una descomposición radix-4 produce alguna eficiencia computacional debido a que el DFT de 4 puntos tiene una conformación tipo mariposa mucho más grande sin multiplicación. Por otro lado, se puede demostrar que usando una descomposición radix con un índice mayor que 4 no resulta en una reducción significativa en la complejidad computacional.

Si se usa el algoritmo FFT radix-4 con decimación en frecuencia para muestras impares del DFT de N puntos, se obtiene los siguientes DFT's de $N/4$ puntos:

$$X(4k+1) = \sum_{n=0}^{N/4-1} \left\{ \left[x(n) - x\left(n + \frac{N}{2}\right) \right] - j \left[x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right) \right] \right\} W_N^n W_{N/4}^{kn} \quad (2.55)$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} \left\{ \left[x(n) - x\left(n + \frac{N}{2}\right) \right] + j \left[x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right) \right] \right\} W_N^{3n} W_{N/4}^{kn}$$

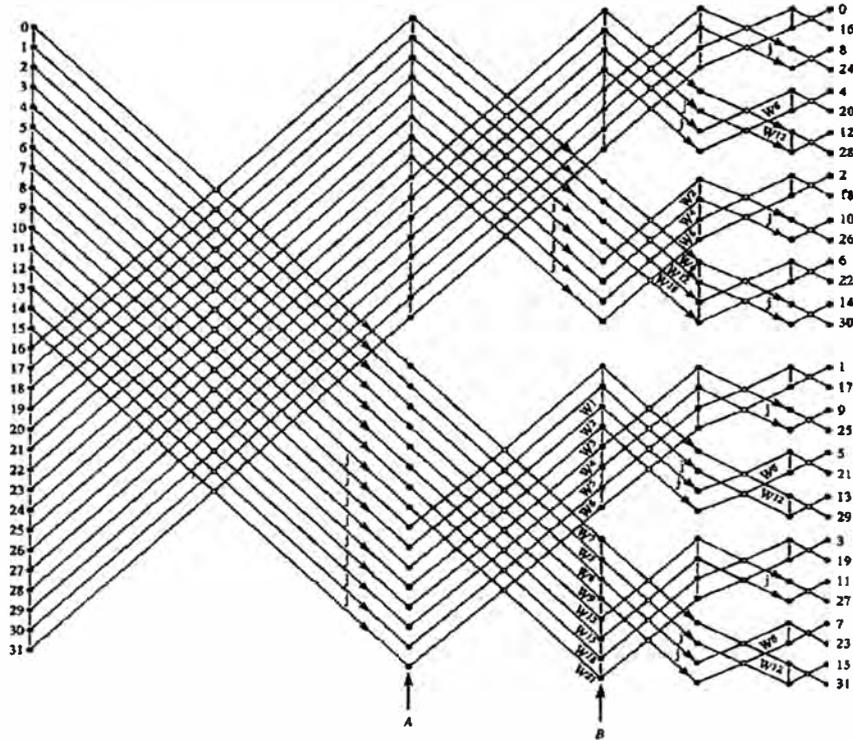


Fig. 2.19.- Algoritmo SFFT con decimación en frecuencia de 32 puntos

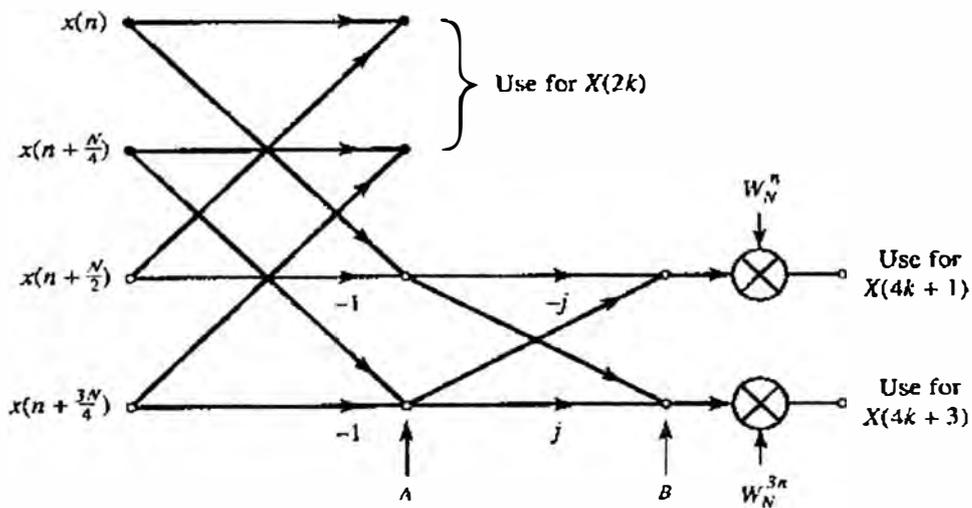


Fig. 2.20.- Algoritmo SFFT computación mariposa

En la siguiente tabla, se muestra un resumen de los resultados para el cálculo de la FFT de hasta 1024 puntos que brindaría aplicar la metodología Radix- n , para los índices de 2, 4 y 8, así como la metodología del Split Radix.

En ésta se pueden comprobar las deducciones a que se refirieron páginas arriba.

TABLA 2.1.- Numero de Multiplicaciones reales no triviales y sumas para computar un DFT complejo de N -puntos

N	Multiplicaciones Reales				Sumas Reales			
	Radix-2	Radix-4	Radix-8	Split Radix	Radix-2	Radix-4	Radix-8	Split Radix
16	24	20		20	152	148		148
32	88			68	408			388
64	264	208	204	196	1032	976	972	964
128	72			516	2054			2308
256	1800	1392		1284	5896	5488		5380
512	4360		3204	3076	13566		12420	12292
1024	10248	7856		7172	30728	28336		27652

2.3.7. Implementación de la transformada rápida de Fourier (Algoritmos)

a. El algoritmo de Tukey-Cooley:

El algoritmo FFT desarrollado por Tukey-Cooley es uno de los más ampliamente usados. También es un algoritmo basado en la estrategia “divide y conquistarás”, pues a través de fraccionar una DTF (Transformada Discreta de Fourier) compuesta de un tamaño $n = n_1 n_2$, se obtienen DFT's de tamaños n_1 y n_2 , junto con $O(n)$ multiplicaciones de raíces de unidad complejas, tradicionalmente llamados twiddle factors.

Este método y la idea general de esta FFT, fue popularizada por una publicación de J.W. Cooley y J.W. Tukey en 1995, pero se descubrió posteriormente que estos dos autores, habían reinventado independientemente un algoritmo conocido por Carl Friedrich Gauss alrededor de 1805.

El uso más conocido de los algoritmos de Tukey-Cooley es dividir la transformada en dos piezas de $n/2$ de tamaño, y está limitado por lo tanto a un tamaño de potencias de dos, pero cualquier factorización puede ser usada en general. Estas son llamados los casos de radix-2 y mixed-radix, respectivamente (hay otras variantes más). A pesar de que la idea básica es recursiva, las implementaciones más tradicionales reacomoda el algoritmo para evitar la recursión explícita. También, dado que el algoritmo de Tukey-

Cooley rompe el DTF en pequeños DTF's, este puede ser combinado arbitrariamente con cualquier otro algoritmo par la DTF, como los descritos abajo.

b. Otros algoritmos FFT:

Existen otros algoritmos FFT distintos a los de Tukey-Cooley. Por ejemplo, para $n = n_1 n_2$, siendo n_1 y n_2 números primos relativos, uno puede usar el algoritmo del Factor Primo (PFA) de Good-Thomas, basado en el teorema del recuerdo chino, para factorizar el DFT similar al de Tukey-Cooley pero sin los factores twiddle.

Se tiene también el algoritmo de Rader-Brenner que consiste en una factorización similar a la de Tukey-Cooley, pero con factores twiddle puramente imaginarios, reduciendo las multiplicaciones a costa de incrementar las sumas y reducir la estabilidad numérica. Algoritmos que factorizan recursivamente la DFT en operaciones más pequeñas, incluyen los algoritmos de Bruun y SFT (Los algoritmos de Rader-Brenner y QFT fueron propuestos para tamaños de potencias de dos, pero es posible que estos puedan ser adaptados en un número general n . El algoritmo de Bruun se aplica a tamaños compuestos pares arbitrarios). El algoritmo de Bruun, en particular, está basado en la interpretación de la FFT como una factorización recursiva del polinomio $z^n - 1$, en polinomios con coeficientes reales de la forma $z^m - 1$ y $z^{2m} + az^m + 1$.

Otro punto de vista polinomial es el explotado por el algoritmo Winograd, el cual factoriza el polinomio $z^n - 1$, en polinomios ciclotómicos (estos a menudo tienen coeficientes de 1, 0, o -1), y por lo tanto requiere pocas multiplicaciones (si requiriera alguna); así, Winograd puede ser usado para obtener mínimas multiplicaciones de FFT's y es a menudo usada para encontrar algoritmos eficientes para factores pequeños. En particular, Winograd también hace uso del PFA como también un algoritmo de Arder para FFT de tamaños primos. El algoritmo de Rader, explotando la existencia de un generador para el grupo multiplicativo módulo primo n , expresa una DFT de tamaño primo n como un convolución cíclica de tamaño $n-1$, el cual puede entonces ser iterada por un par ordinario de FFT's vía el teorema de convolución (a pesar de que Winograd usa otros métodos de convolución). Otra FFT de tamaño primo es el debido a L. I. Bluestein, y es algunas veces llamado algoritmo chirp-z; este también re-expresa la DFT como una convolución, pero esta vez del mismo tamaño, vía la identidad $jk = - (j - k)^2/2 + j^2/2 + k^2/2$ (la cual puede ser llenada por ceros a una potencia de dos y ser evaluada por la FFT de radix-2 Cooley-Tukey, por ejemplo).

CAPÍTULO III

METODOLOGÍA PARA LA SOLUCIÓN DEL PROBLEMA: IMPLEMENTACIÓN DEL ANALIZADOR DE ESPECTROS DIGITAL

3.1. Alternativas de solución

Con el objetivo de lograr el desarrollo de un instrumento analizador de espectros digital de bajo costo y buena funcionalidad para el laboratorio, se pueden considerar entre varias otras, las siguientes principales alternativas:

3.1.1. Analizador Digital Desarrollado en Hardware.-

Una opción económica en hardware para el desarrollo de un analizador de espectros de laboratorio, podría estar basado en el uso de una PC, con el cual se podrían combinar las capacidades gráficas de éste y para el procesamiento digital de señales usar un chip dedicado, como el Texas Instruments TMS320C25 que es un DSP de alta potencia en el procesamiento numérico, junto con el desarrollo de la electrónica de soporte para este. Se requeriría construir un bloque de adquisición, para tomar la señal analógica, digitalizarla y calcular su espectro utilizando el algoritmo de FFT, y mostrar en la pantalla de la PC el espectro en frecuencia obtenido (4).

Un diagrama de bloques conceptual del analizador de espectros es mostrado en la figura 3.1. El analizador tendría la función de ejecutar por ejemplo, la FFT basado en el algoritmo Radix 2 de N -puntos sobre los datos de una de las señales obtenidas en la entrada, calcular luego la densidad de potencia espectral y transmitir luego a la PC vía una interfase RS232C. Se podría elaborar un programa en el lenguaje C para recibir y mostrar en pantalla el espectro.

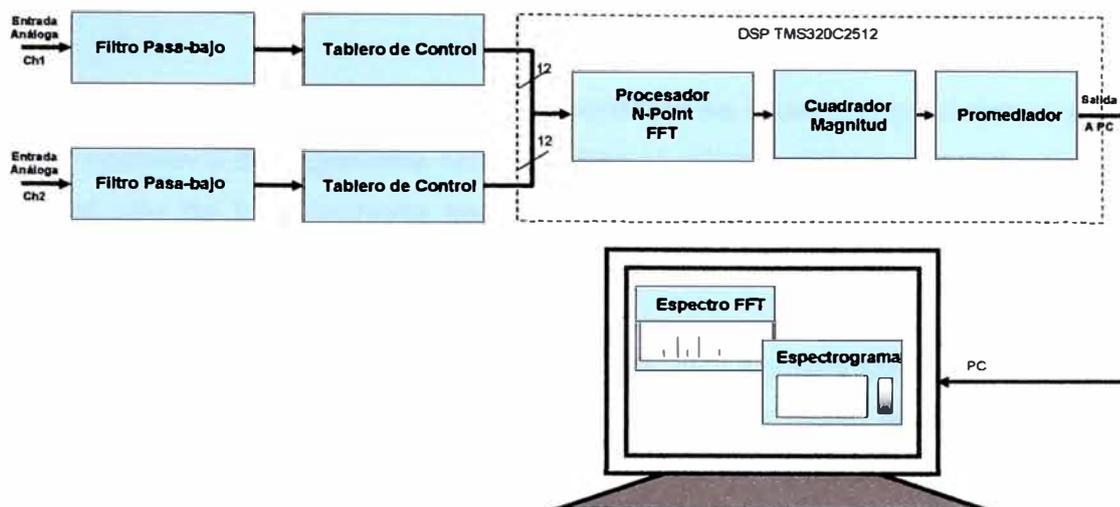


Fig. 3.1.- Diagrama de Bloques Analizador de Espectros HW

La ventaja de esta solución es que toda la potencia en el procesamiento numérico para el cálculo de la FFT es realizada en el TMS320C25 en forma dedicada, lo cual mejora en mucho la eficiencia del instrumento, y para mostrar los resultados se tiene la interacción con la PC utilizando funciones típicas.

3.1.2. Analizador Digital Desarrollado en Software.-

Bajo las consideraciones que el desarrollo implique un bajo costo, se plantea la alternativa de desarrollar el instrumento en software, donde no se incluye ningún módulo físico o hardware (ni para la adquisición de la señal, ni para discretizarla, ni para procesarla). Se trata de elegir un software aplicativo que tome la señal a procesar de alguna manera, y calcule el FFT y muestre el espectro de la señal. La ventaja de este método, es que se ahorraría todo el costo de la implementación física, que conlleva complicaciones, tanto para desarrollarla, así como para calibrarla, y escapa los propósitos de este trabajo.

3.2. Solución al problema

Dentro de estas son muchas las alternativas, pero se plantea como parte del presente trabajo, el uso del aplicativo de ingeniería MatLab, considerando las funciones avanzadas que poseen para el análisis de señales, así como el cómputo de la Transformada Rápida de Fourier (FFT).

3.3. Concepción práctica del sistema

El diseño del instrumento digital a construir se basa en aplicaciones programadas sobre el aplicativo de ingeniería MatLab. Por el mismo objetivo de este documento, se plantea el uso de transformada rápida de Fourier (FFT-Fast Fourier Transform) como herramienta para el análisis espectral de las señales.

El desarrollo se conceptúa en dos partes, una primera etapa del mismo, incluye el diseño e implementación del instrumento base, y una segunda etapa, donde se orienta el uso del instrumento a aplicaciones prácticas, tal como se puede explicar en seguida:

3.3.1. Primera etapa.- Construcción del instrumento Básico y calibración.-

Comprende el diseño de un instrumento de medición, básicamente un espectrógrafo, el que se ha desarrollado en base a una secuencia de scripts en el aplicativo de ingeniería MatLab. Este desarrollo permite la construcción del instrumento básico, que permita la calibración del mismo, a través de lecturas de funciones conocidas (en este caso, una señal senoidal), con lo cual se puede verificar el correcto comportamiento del instrumento para las condiciones establecidas.

Para este mismo efecto, se ha desarrollado una interfaz gráfica interactiva en PC, para que el usuario pueda seleccionar los siguientes parámetros: (1) La frecuencia de la señal senoidal base, (2) El número de períodos que se desea analizar y visualizar, (3) El número de puntos de la FFT, con la cual se pueda elegir una mayor precisión en el cálculo de la transformada rápida de Fourier de la señal base, (4) La frecuencia de Muestreo, de forma que se vea el efecto en la respuesta en frecuencia, y (5) El nivel de resolución, que permita visualizar con mayor o menor precisión la frecuencia en el módulo del frecuencímetro digital.

A través de la invocación de los "CallBack", se actualizan las condiciones de las funciones, y se refrescan los datos en pantalla.

En general, el dispositivo también debería incluir la opción de capturar señal y aplicar sobre el, el tratamiento aplicado a la señal de prueba, de forma que se puedan observar el espectro en frecuencia.

3.3.2. Sistema de Aplicación.-

Para el caso del presente estudio, se presentan tres casos de análisis de señales para su análisis de espectros usando en módulo en MatLab. El diseño del analizador de espectros está orientado al tratamiento de las señales de baja frecuencia.

Para el presente caso, se está considerando las siguientes señales de baja frecuencia: (1) Electro-encefalograma, (2) Electro-cardiograma y (3) Señales sismográficas; las cuales serán analizadas en el instrumento, para graficar sus respuestas en frecuencia y su comportamiento en el espectrograma, luego se da una interpretación del mismo.

Dadas las limitaciones del alcance de este estudio, no se considera la implementación del modulo hardware que actúe como interfaz para la adquisición de las señales arriba indicadas, simplemente se considera la adquisición a nivel de lectura de datos almacenados en archivo electrónico conteniendo la información de las señales en disco de la PC, esto a través de comando de lecturas del MatLab. Esto requiere la adaptación previa de la información, y considerando las limitaciones de análisis de un solo canal, se seleccionan información de canales específicos para el análisis.

3.4. Características y especificaciones del analizador de espectros

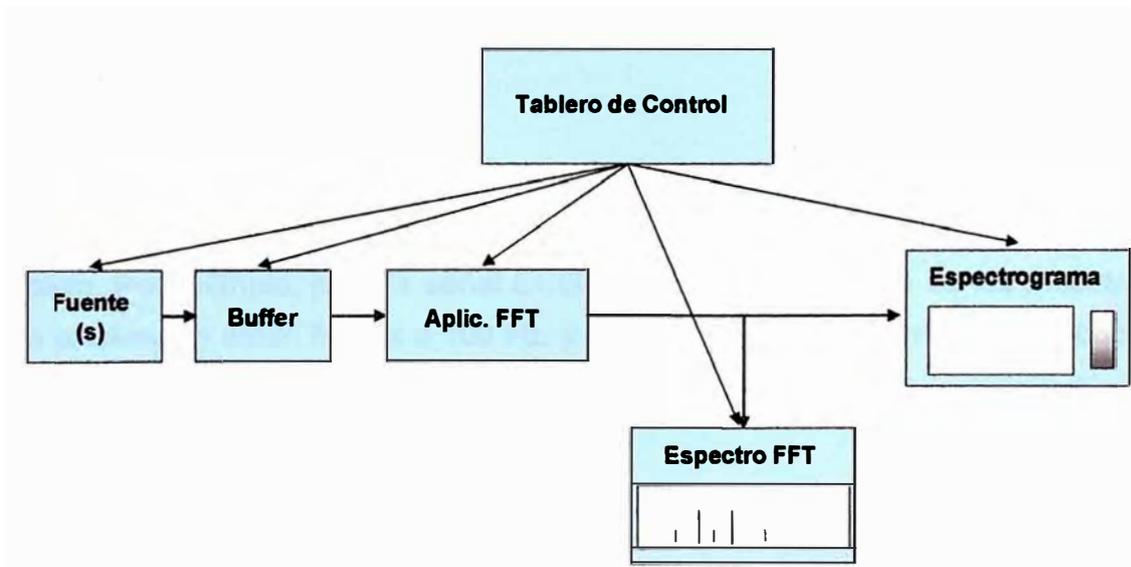


Fig. 3.2. Diagrama de Bloques del Sistema

3.4.1. Descripción del Sistema.-

a.- Fuente – Señal de Entrada.-

En general, el elemento fuente podría ser un módulo de adquisición de datos adaptado a la PC. En general si fuera señal de audio, los datos podrían ser ingresados directamente a través de un micrófono, o el conector de entrada de la tarjeta de sonidos de la PC. Para otras señales con otras frecuencias, se podrán utilizar tarjetas especializadas incorporadas a la PC, u otros módulos externos, que se adapten vía cable paralelo o puerto USB a la PC.

En el caso de la realización presente, simplemente se está simulando la adquisición de los datos de las señales, las cuales se están adquiridas de los registros en archivo electrónico tipo "txt", las cuales luego son leídas y cargadas en variables específicas dentro del aplicativo, para luego ser analizadas.

b.- Buffer.-

En un caso real, requeriríamos de un elemento que haga las funciones de buffer, para retener la información discreta de la señal procedente de la fuente. En el caso particular que nos compete, la función la hace la memoria de la misma PC que en una variable arreglo matricial, guarda valores discretizados de la señal entrante. Para esto se carga un número de muestras general y se almacenan en la variable Si que hace las veces de buffer.

Este tamaño del buffer está muy relacionado con la frecuencia de muestreo que se esté usando. En el caso particular del instrumento, F_s es definido por la señal ya discretizada. Por ejemplo, para la señal ECG, la frecuencia definida de las muestras que se están analizando están fijadas a 100 Hz, y este valor es el definido para el modo ECG.

c.- Módulo Control y Ajustes.-

Permite el ajuste de diversos parámetros, que definen entre otras cosas, los tamaños de muestras, la cantidad de puntos de evaluación de la FFT, ventana de análisis, controles de acercamiento (zoom) de los diversos cuadros de presentación de señales, etc. .

d.- Módulo de Aplicación del Espectro FFT.-

Se está planteando el tratamiento de la señal a través de la aplicación de la Transformada Rápida de Fourier directa (FFT), para graficar el resultado del análisis en el diagrama de densidad espectral en función de la frecuencia.

e.- Módulo Espectrograma.-

Como otra herramienta adicional para analizar el dominio de la frecuencia de las señales de entrada, el instrumento también incorpora un módulo para medir el espectrograma, que implemente la función correspondiente del MatLab y previamente se usa la función del filtro Hanning.

3.5. Soporte de software

3.5.1. Aspectos de programación del software base

a.- Diseño.-

El sistema se conceptúa de una forma modular, de forma que, a la interacción del usuario con la interfaz gráfica, pueda invocarse la función principal que efectúa el análisis espectro-gráfico de la señal definida. Toda la programación ha sido elaborada en base a scripts soportadas en el aplicativo MatLab, tomando funciones básicas existentes en la librería, y desarrollando los módulos en base a ellas.

3.5.2. Implementación de algoritmos – Programa de calibración

a.- Módulo Interactivo con PC.-

Este modulo es el encargado de capturar los datos seleccionados en la pantalla interactiva, y con cuyos datos luego invoca al módulo del espectrógrafo digital.

La función principalmente usada son los “CallBack” en los módulos del MatLab “uicontrol”. Se han desarrollado sub-módulos para la captura de los valores numéricos cada uno de los parámetros considerados, y dentro de ellos se invoca, como yo se ha

indicado, al módulo principal del instrumento. El script correspondiente a este módulo, puede ser hallado en el Anexo C.1.1

b.- Módulo Espectrógrafo (Espectro FFT).-

En este módulo de programación se considera: (1) La conformación de la señal base, en función de la frecuencia previamente determinada en el módulo de interfaz usuario, (2) La graficación de la misma, esto en función al número de muestras seleccionado, la frecuencia base y la frecuencia de muestreo; donde este último parámetro determina la precisión en la representación de la señal, puesto que a mayor frecuencia de muestreo, se podrá apreciar una representación más cercana a la señal original. (3) La graficación de la respuesta en frecuencia, en lo que constituye el espectrógrafo digital. Cabe indicar que sobre el interactúa el parámetro de resolución. El valor por defecto es 100%, pero puede reducirse hasta un mínimo, y cuanto menor sea el valor seleccionado, este permitirá ver con más acercamiento la respuesta en frecuencia. (4) La graficación de la respuesta en frecuencia en un espectrograma; es decir, visualizar la respuesta en frecuencia en función del tiempo, la frecuencia y la potencia de dicha señal.

c.- Módulo aplicativo - lectura y procesamiento de Señal.-

El script correspondiente a este módulo, puede ser hallado en el Anexo C.1.2.

3.5.3. Implementación de algoritmos – Programa de Aplicación

Esta etapa del desarrollo, contempla la elaboración del script de la parte aplicativo del sistema. Como ya se había indicado, se estaría orientando esta aplicación al análisis de señales de baja frecuencia, en el caso particular señales biológicas como las adquiridas de cardiogramas, encefalogramas, y además, señales sísmicas. Sin embargo, la aplicación del instrumento es más amplia y general, y podría analizar señales de otros tipos, sean señales de audio, u otras de frecuencia mayor, para lo cual se requeriría simplemente, implementar el módulo de adquisición o si ya está adquirida la señal, elaborar el módulo de adecuación de la información de la señal, para que esta pueda ser leída correctamente por el aplicativo.

Al igual que en la etapa anterior, el desarrollo se plantea en dos módulos principales: el módulo interactivo con PC y el módulo analizador de espectros.

a.- Módulo Interactivo con PC.-

Es similar que el módulo interactivo de la primera etapa, pero por razones obvias, ya no se consideran como parámetros la frecuencia de señal base, puesto que esta será leída de un archivo de datos en el segundo módulo, así como tampoco se considera como dato de entrada la frecuencia de muestreo, pues esta está ya predefinida. Se mantienen los sub-módulos que permiten la captura del número de muestras de la señal (N_s), el número de puntos de evaluación de la FFT (N), y el nivel de resolución, similar al desarrollado en la etapa primera. El script correspondiente a este módulo puede ser hallado en el Anexo C.2.1

b.- Módulo Espectrógrafo del ECG-EEG-SISM.-

En este módulo de programación se considera: (1) La captura de la información de los registros de señales reales de electrocardiograma, electroencefalograma, o señal sísmica, las cuales están grabadas en archivos en PC, y los que luego se leen y almacenan en una variable en el programa, (2) La graficación de la misma, esto en función al número de muestras seleccionado, la frecuencia base y la frecuencia de muestreo; donde este último parámetro determina la precisión en la representación de la señal, puesto que a mayor frecuencia de muestreo, se podrá apreciar una representación más cercana a la señal original. (3) La graficación de la respuesta en frecuencia, en lo que constituye el espectrógrafo digital. Cabe indicar que sobre el interactúa el parámetro de resolución. El valor por defecto es 100%, pero puede reducirse hasta un mínimo, y cuanto menor sea el valor del valor seleccionado, este permitirá ver con más acercamiento la respuesta en frecuencia. (4) La graficación de la respuesta en frecuencia en un espectrograma; es decir, visualizar la respuesta en frecuencia en función del tiempo, la frecuencia y la potencia de dicha señal.

Para la selección de la opción ECG, EEG o SISM, simplemente se activa la operación fopen que abre el archivo correspondiente a cada una de las opciones. Para esto los archivos con la data del ECG, EEG y SISM, se tienen almacenadas en el directorio "work" dentro del directorio del MatLab V.6.5. El script correspondiente a este módulo puede ser hallado en el Anexo C.2.2.

CAPÍTULO IV

ANÁLISIS Y PRESENTACIÓN DE RESULTADOS

4.1. Resultados del analizador de espectros – Módulo de calibración

Ejecución y Resultados de la simulación.- Ejecutamos el script con los siguientes parámetros:

Número de períodos se señal =20

Frecuencia de Señal = 200 Hz

Frecuencia de muestreo= 8024 Hz

Número de puntos evaluación FFT=1024

Resolución = 11,6%

Con lo cual, tenemos los siguientes resultados:

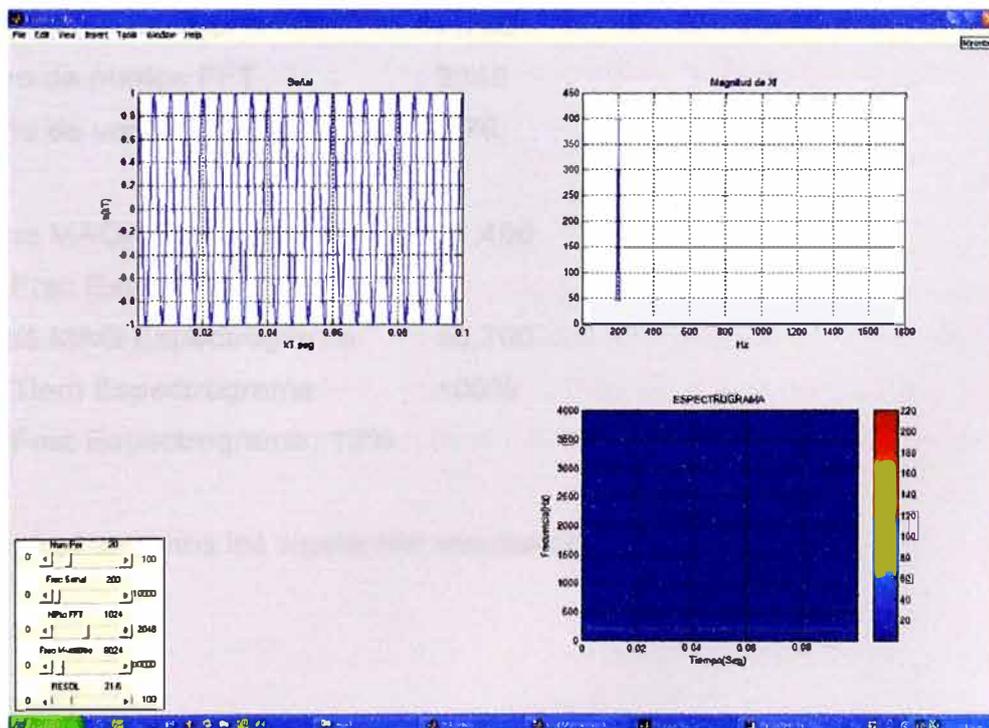


Fig. 4.1.- Resultados obtenidos con módulo de calibración

- (1) El diagrama de la señal generada, que en este caso está dada por la señal senoidal del 200 Hz, según lo escogido en el control de barras.
- (2) El diagrama de la señal en el dominio de la frecuencia, aplicado la resolución a 11,6%. Igualmente, esta se puede identificar en el espectrograma, pero con baja intensidad, por el efecto de la escala.

Adicionalmente a la prueba de calibración, que se pretendió hacer con el primer desarrollo del aplicativo para este efecto, se puede incluir los resultados en el aplicativo final, al que se le bloquea los módulos de lectura de archivos para la adquisición de datos, y se incluye unas líneas en el script para la conformación de señales senoidales con:

$$S = \cos(2\pi t f) + \cos(2\pi t^2 f) + \cos(2\pi t^4 f); \text{ y } f = 100\text{Hz.}$$

Los resultados son los que se muestran más abajo.

Ejecución y Resultados de la simulación.-

Ejecutamos el script con los siguientes parámetros:

Menú Opción	: PATRON
Número de Muestras	: 7,782
Número de puntos FFT	: 2048
Tamaño de ventana análisis	: 2176
Shift	: 0
Ventana MAGX Espectrógrafo	: 11,400
Zoom Frec Espectrograma	: 13%
Ventana MAG Espectrograma	: 30,700
Zoom Tiem Espectrograma	: 100%
Zoom Frec Espectrograma:	12%

Con lo cual tenemos los siguientes resultados:

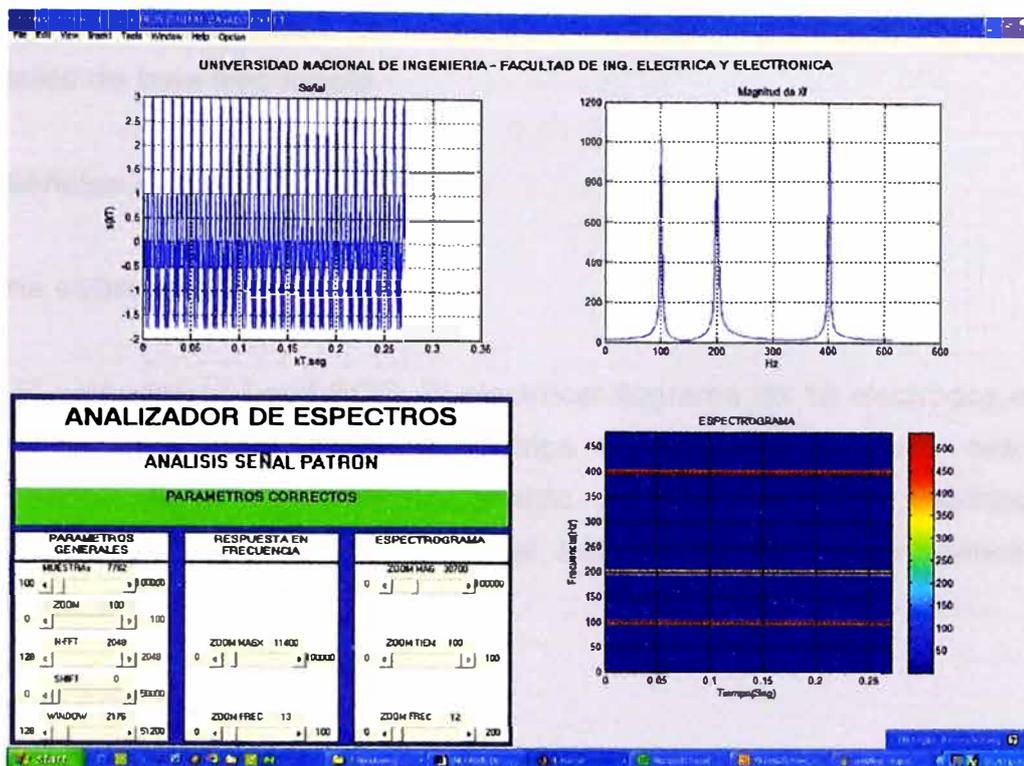


Fig. 4.2.- Información relativa al análisis de las Señales patrón para calibración de instrumento

- (1) El diagrama de la señal generada, que en este caso está dada por una composición de señales senoidales a 100, 200 y 400 Hz, se utiliza la frecuencia de muestreo de 8,000 Hz.
- (2) El diagrama de la señal en el dominio de la frecuencia, luego de haber ajustado el control de zoom en la escala de la frecuencia del espectrómetro al 13%, presenta claramente la respuesta en frecuencia según el valor esperado.
- (3) El diagrama de la señal en el Espectrograma. Al ser la señal senoidal pura, se puede apreciar la intensidad de potencia a los 100, 200 y 400 Hz mantenida constante durante todo el tiempo de la evaluación.

De los resultados mostrados, se puede apreciar la versatilidad de las funciones del MatLab para creación de aplicaciones finales de diverso tipo, en este caso, el analizador de espectros. Se puede apreciar la capacidad para poder efectuar el análisis espectral, basado en funciones ya existentes para el tratamiento digital de las señales, como es la Transformada Rápida de Fourier.

4.2. Resultados del analizador de espectros – Modulo aplicación: Adquisición de señales de baja frecuencia

4.2.1. Señales de electrocardiograma

a. Teoría sobre las Señales ECG.-

El estándar 12 Lead ECG: El electrocardiograma de 12 electrodos es una representación de la actividad eléctrica del corazón registrada desde los electrodos de la superficie del cuerpo. A continuación se describen los componentes básicos del ECG y el sistema de electrodos usados para registrar el ECG (10).

Ondas ECG y sus Intervalos.-

La figura 4.3 ilustra las ondas ECG y sus intervalos en el tiempo, así como, las medidas estándar de tiempo y voltaje medidos en el ECG.

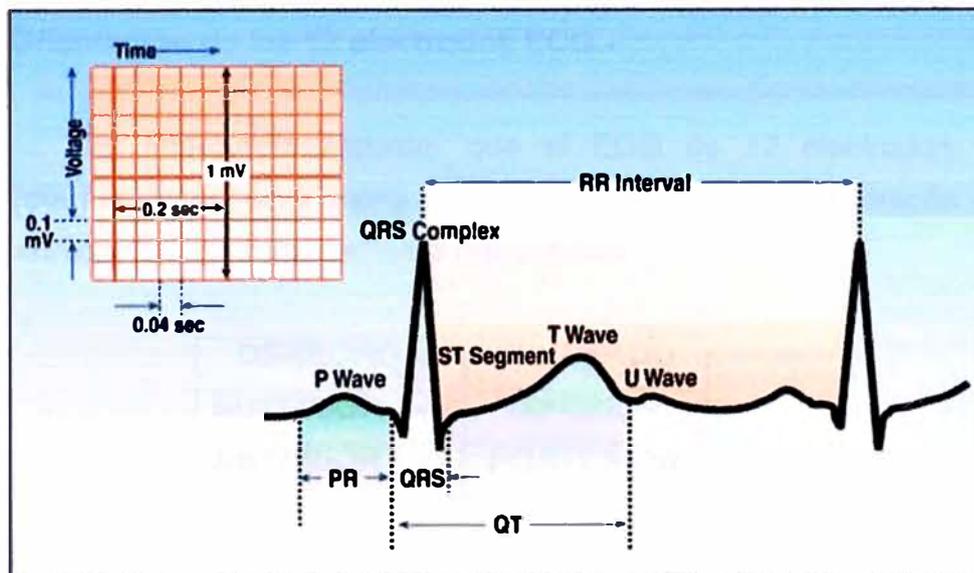


Figura 4.3.- Diagrama estándar 12 Lead ECG

Fuente: http://medstat.med.utah.edu/kw/ecg/ecg_outline/

Donde:

- Onda P: Representa la activación secuencial (depolarización) de las atrias izquierda y derecha.

- Complejo QRS: Depolarización ventricular derecha e izquierda (normalmente los ventrículos son activados simultáneamente).
- Onda ST-T: Repolarización ventricular.
- Onda U: Se indica que el origen de esta onda no es clara, pero probablemente representa las depolarizaciones posteriores en los ventrículos.
- Intervalo PR: Intervalo de tiempo desde el comienzo de la depolarización atrial (Onda P) al comienzo de la depolarización ventricular (Complejo QRS).
- Duración del QRS: Duración de la depolarización de músculo ventricular.
- Intervalo QT: Duración de la depolarización ventricular y la repolarización.
- Intervalo RR: Duración del ciclo cardiaco ventricular (un indicador de la velocidad ventricular).
- Intervalo PP: Duración del ciclo atrial (un indicador de la velocidad atrial).

Orientación de los 12 electrodos ECG.-

Es importante recordar que el ECG de 12 electrodos provee información espacial acerca de la actividad eléctrica del corazón en tres direcciones aproximadamente ortogonales:

DERECHO	↔	IZQUIERDO
SUPERIOR	↔	INFERIOR
ANTERIOR	↔	POSTERIOR

Cada una de los 12 electrodos representa una orientación particular en el espacio, como la indicada abajo:

- RA: Brazo derecho (right arm).
- LA: Brazo izquierdo (left arm).
- LF: Pie izquierdo (left foot)

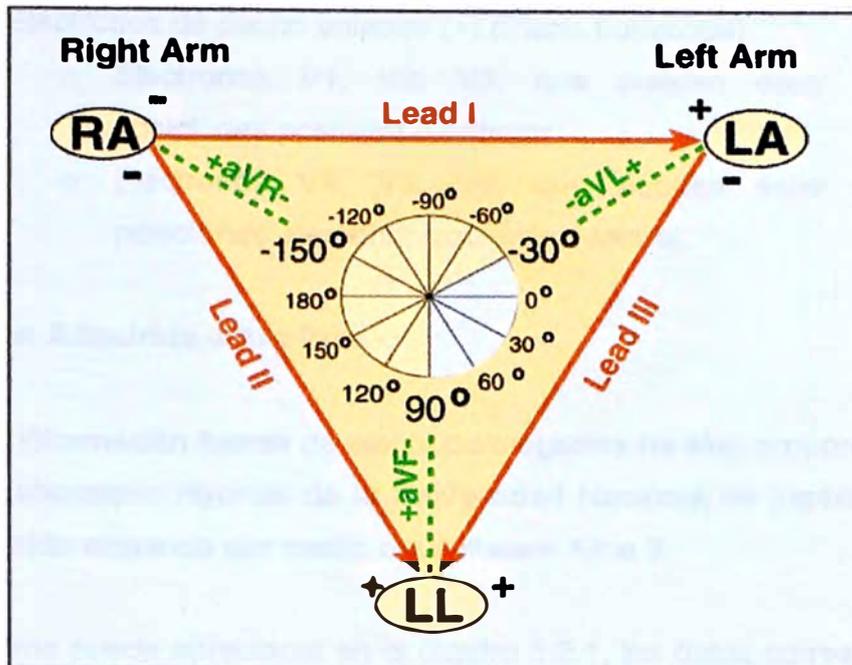


Figura 4.4.- Triángulo de Einthoven

Fuente: http://medstat.med.utah.edu/kw/ecg/ecg_outline/

Donde: Cada uno de los 6 electrodos del plano frontal tienen una orientación positiva y negativa (Indicadas por los signos '+', y '-').

Descripción de los Electrodo:

- Electrodo de extremidad bipolar:
 - Electrodo I: RA (-) a LA(+), que puede estar en posición derecha, izquierda o lateral.
 - Electrodo II: RA (-) a LF(+), que puede estar en posición superior o inferior).
 - Electrodo III: LA (-) a LF (+), que puede estar en posición superior o inferior).

- Electrodo de extremidad de incremento unipolar (plano frontal):
 - Electrodo aVR: RA (+) a [LA & LF](+), en el sentido derecho.
 - Electrodo aVL: LA (+) a [RA & LF](-), en el sentido izquierdo.
 - Electrodo aVF: LF (+) a [RA & LA](-), en el sentido inferior.

- Electrodo de pecho unipolar (+) (Plano horizontal)
 - Electrodo V1, V2, V3, que pueden estar en las posiciones posterior o anterior.
 - Electrodo V4, V5, V6, que pueden estar en las posiciones: derecha, izquierda o lateral.

b. Fuente de Data Adquirida a Analizar.-

La información fuente de electrocardiograma ha sido proporcionada por el Laboratorio Hypnos de la Universidad Nacional de Ingeniería, la cual ha sido adquirida por medio del software Alice 3.

Como puede apreciarse en el cuadro 5.2.1, los datos corresponden a señales de electrocardiograma tomadas de una persona de sexo femenino, de 18 años de edad, tomadas el día 18 de Mayo del 2004, para lo cual se utilizó electrodo de oro. La frecuencia de muestreo aplicada a las señales es de 100 Hz y ha sido almacenada en archivo en formato tipo texto.

```

HeadECGPac0 - Bloc de notas
Archivo Edición Formato Ver Ayuda
# These signals(data) are part of a set of signals:
# ECG, EEG, EMG, EOG, Resp1 y Ronq1 extracted
# from patients in Sleep Laboratory of Hypnos, using Alice 3
Acquisition Software

Name : Paciente0
Sex : Female
Age : 18 years
Start_date : 05/18/04
Start_time : 22:41:00
End_time : 07:12:59
Signals : Time_offset, ECG, RR
Electrodes : Surface_Electrodes_Gold

File          Rate    Samples
ECGPac01.txt  100    360000
ECGPac02.txt  100    360000
ECGPac03.txt  100    360000
ECGPac04.txt  100    360000
ECGPac05.txt  100    360000
ECGPac06.txt  100    360000
ECGPac07.txt  100    360000
ECGPac08.txt  100    360000
ECGPac09.txt  100    192000

# Rate : sampling rate in Hz
# Samples : number of samples in file
  
```

Fig. 4.5 Información relativa a la adquisición de Señales ECG

Para fines del procesamiento se tomó la muestra del archivo ECGPac03.txt (la cual contiene un total de 360,000 muestras de la señal), y a fin de adecuarla y almacenarla en la variable, se volcó la información en una etapa intermedia con el aplicativo de excel, y se eliminaron los datos de tiempo que resultan innecesarios, dado que se tiene como dato la frecuencia de muestreo de 100 Hz. definida que sea fácil de leer por el aplicativo Analizador de Espectros

La desventaja de usar el excel para esto es que trunca la información hasta los 65536 muestras, que es el número máximo de celdas de este aplicativo del MS Office. La información de la señal fue recogida en el archivo 'ECGPacO3full.txt'.

Ejecución y Resultados de la simulación.-

Ejecutamos el script con los siguientes parámetros:

Número de Muestras	: 31,500
Zoom tiempo	: 100%
Número de puntos FFT	: 1024
Tamaño de ventana análisis	: 3,456
Shift	: 100
Ventana MAGX Espectrógrafo	: 53,000
Zoom Frec Espectrografo	: 7,7%
Ventana MAG Espectrograma	: 11,300
Zoom Tiem Espectrograma	: 100%
Zoom Frec Espectrograma:	11%
Archivo de datos fuente	: ' ECGPacO3full.txt '

Con lo cual, tenemos los siguientes resultados en la figura 4.6:

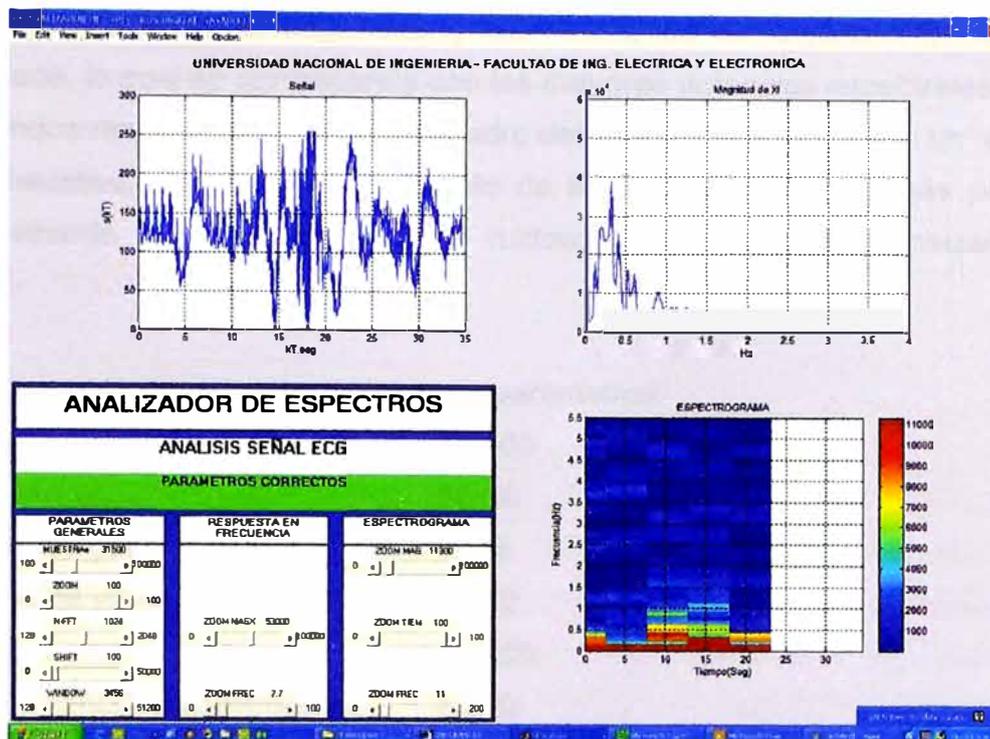


Fig. 4.6.- Resultados del análisis de señal ECG

- (1) El diagrama de la señal ingresada desde el archivo y volcada en la variable de señal S, para un total de 31,500 muestras. Se observa un alto componente continuo o de frecuencia cero alrededor de las 140 unidades de amplitud.
- (2) El diagrama de la señal en el dominio de la frecuencia, aplicado el control del zoom del eje de la frecuencia al 7,7%, se puede apreciar la alta actividad espectrográfica para frecuencias menores de 1 Hz, específicamente, el pico más alto está alrededor del 0,3 Hz.
- (3) El diagrama de la señal en el Espectrograma, ajustando los parámetros de control indicados, se aprecia la misma actividad espectral ya indicada en el punto anterior, es decir, mayor potencia de la señal en el rango menor a 1 Hz. La observación de mosaicos de colores un poco gruesa, que no permite visualización de una forma más fina la actividad espectral, es debido al efecto zoom aplicado.

La ventaja adicional del espectrograma como instrumento de análisis espectral, es que se puede observar además la respuesta en frecuencia en el tiempo. Es así, que el caso analizado, en el diagrama de señal, se observa una

actividad fuerte a los 10 segundos, y otra vez a los 40 segundos de la señal muestreada, lo cual se corresponde con las mayores potencias espectrales a 10 y 40 segundos respectivamente, en el cuadro del espectrograma. Para un análisis más exhaustivo, tomemos un segmento de la señal muestreada más pura, es decir, evitando coger los segmentos ruidosos, para lo cual desplazamos la ventana.

Ejecutamos el script con los siguientes parámetros:

Número de Muestras : 31,500
 Zoom tiempo : 100%
 Número de puntos FFT : 1024
 Tamaño de ventana análisis : 1,152
 Shift : 16,500
 Ventana MAGX Espectrógrafo : 3,200
 Zoom Frec Espectrografo : 8,3%
 Ventana MAG Espectrograma : 1,400
 Zoom Tiem Espectrograma : 100%
 Zoom Frec Espectrograma: 11%
 Archivo de datos fuente : ' ECGPacO3full.txt '

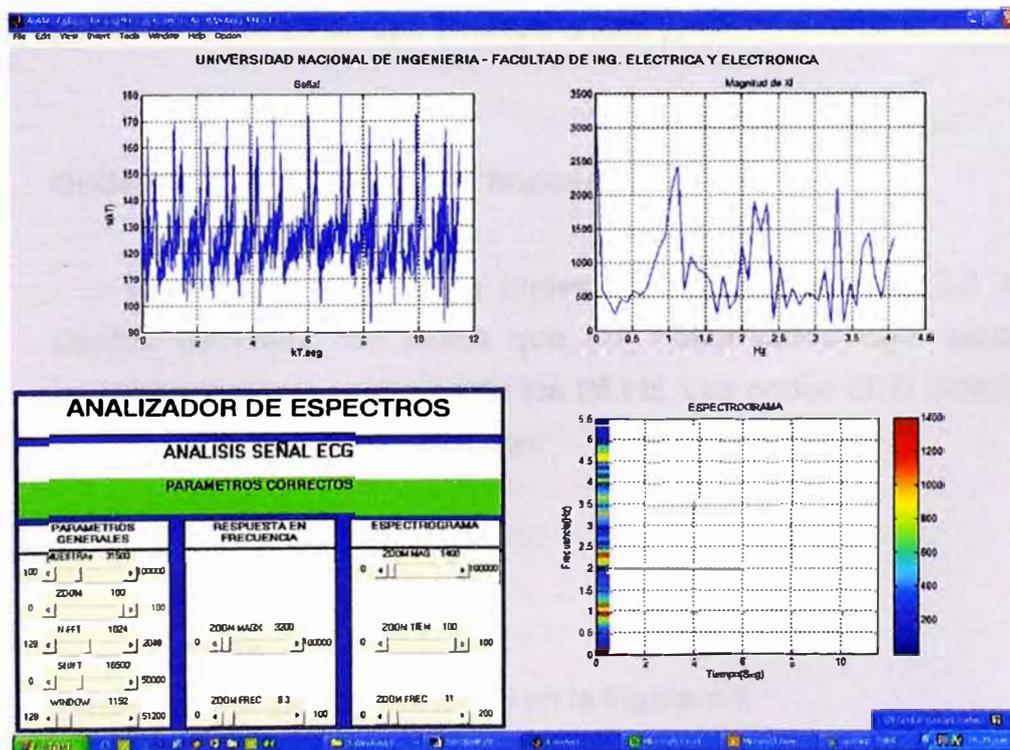


Fig. 4.7.- Resultados del análisis de señal ECG

El segmento tomado muestra señales más puras, por lo que el análisis espectral también es más puro. El pico más notorio, tiene que ver con la pulsación del corazón y se aprecia que es aproximadamente a 1,2 Hz. También se nota componentes espectrales importantes, a 2,3 y 3,3 Hz.

4.2.2. Señales de electroencefalograma

a. Teoría sobre las Señales EEG.-

El EEG detecta los impulsos eléctricos en el cerebro (ondas cerebrales, y las registra y grafica. El EEG puede ayudar al diagnóstico de condiciones tales como la epilepsia, tumores cerebrales, daños en el cerebro, parálisis cerebral, condiciones metabólicas por enfermedades en del hígado, riñón, o incluso la muerte cerebral. Esto también puede ayudar a los médicos a encontrar causas de problemas tales como, dolores de cabeza, debilidad, desvanecimiento, mareos, etc.

Las señales generadas por la actividad encefálica (EEG), involucra la interpretación de las formas de onda mayormente por las frecuencias de dichas ondas y en un menor grado por la morfología de las mismas, o por la complejidad de sus diversas ondas (11).

Ondas EEG – Definidas por frecuencia.-

Las frecuencias de las ondas EEG van desde los 0.5 Hertz a cientos de Hertz. Se indica que los instrumentos EEG usualmente muestran las frecuencias hasta los 26 Hz. Las ondas EEG definidas por sus frecuencias básicamente son:

- Alpha
- Beta,
- Theta, y
- Delta

Cuyas morfologías se muestran en la Figura 4.8.

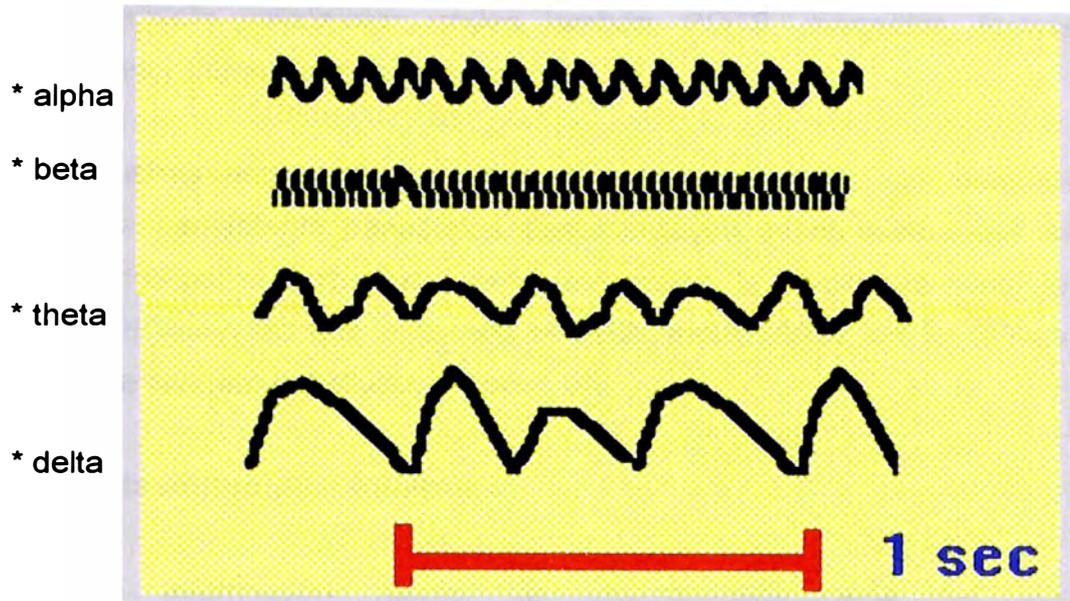


Figura 4.8.- Ondas EEG por su frecuencia

Fuente: http://www.brown.edu/Departments/Clinical_Neurosciences/louis/eegcrs.html

Ondas EEG – Definidas por su morfología.-

Ciertas ondas tienen formas características independientemente de su frecuencia y son reconocibles por sus formas; en otros casos, grupos de ondas tienen apariencias típicas.

Ondas simples que son especialmente conformadas; por ejemplo, ondas largas y agudas, ondas que se elevan rápidamente a un punto y caen rápidamente con una base que es pequeña comparada con la amplitud de la onda.

Algunas formas de onda pueden ser reconocidas por su morfología y estas incluyen dos principales tipos:

1. Ondas especialmente formadas, y
2. Ondas especialmente formadas complejas

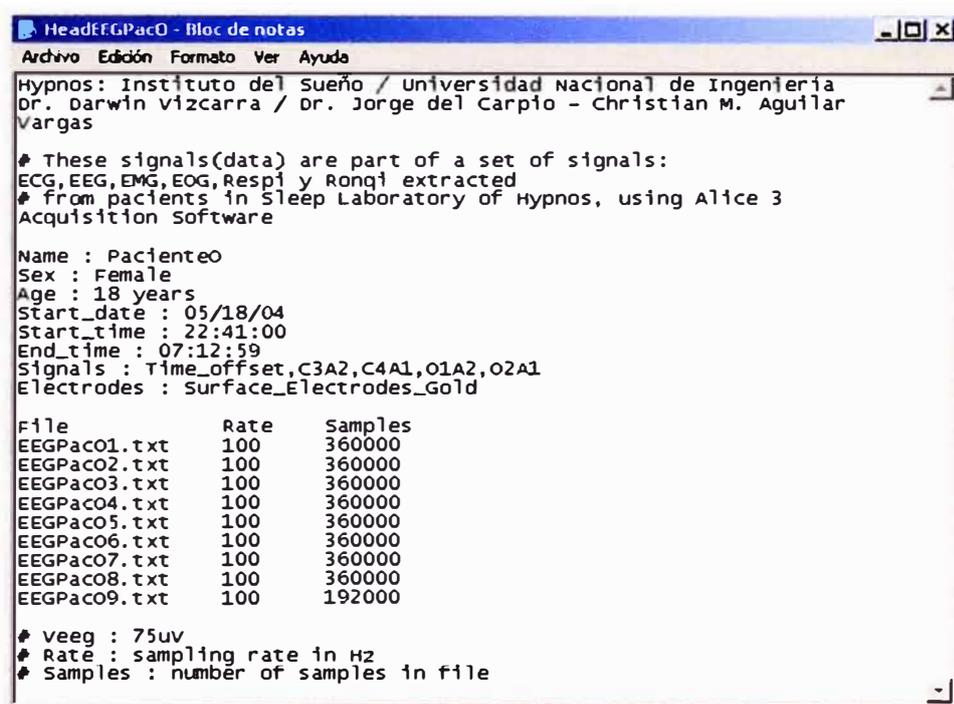
Las variantes normales son diversas ondas o patrones de ondas las cuales son inusuales en apariencia, que a pesar de ello no significa necesariamente una anomalía o enfermedad. Estas ondas pueden ser mal interpretadas. De las ondas más comunes están las de ritmo

mm, variante psicomotor, ondas lambda, POTS, alargadas, formas pulsantes y complejo K, entre otras.

Muchas veces se presentan disturbios causados por defectos técnicos, usualmente transitorios. Estas incluyen casos tales como las generadas por el movimiento de los electrodos, la pérdida de contacto, actividad muscular que oscurece el EEG, movimientos de la cabeza, rascarse el cuero cabelludo, el sudor, etc.

b. Fuente de Data Adquirida a Analizar.-

La información fuente de electroencefalograma ha sido proporcionada por el Laboratorio Hypnos de la Universidad Nacional de Ingeniería, la cual ha sido adquirida por medio del software Alice 3. Como puede apreciarse en la figura 4.9, los datos corresponden a señales de electrocardiograma tomadas de una persona de sexo femenino, de 18 años de edad, tomadas el día 18 de Mayo del 2004, para lo cual se utilizó electrodos de oro en cuatro diferentes canales. La frecuencia de muestreo aplicada a las señales es de 100 Hz y ha sido almacenada en archivo en formato tipo texto.



```

HeadEEGPacO - Bloc de notas
Archivo Edición Formato Ver Ayuda
Hypnos: Instituto del Sueño / universidad Nacional de Ingeniería
Dr. Darwin vizcarra / Dr. Jorge del Carpio - Christian M. Aguilar
Vargas

# These signals(data) are part of a set of signals:
ECG,EEG,EMG,EOG,Respl y Ronq1 extracted
# from patients in Sleep Laboratory of Hypnos, using Alice 3
Acquisition Software

Name : PacienteO
Sex : Female
Age : 18 years
Start_date : 05/18/04
Start_time : 22:41:00
End_time : 07:12:59
Signals : Time_offset,C3A2,C4A1,O1A2,O2A1
Electrodes : Surface_Electrodes_Gold

File          Rate      Samples
EEGPac01.txt  100      360000
EEGPac02.txt  100      360000
EEGPac03.txt  100      360000
EEGPac04.txt  100      360000
EEGPac05.txt  100      360000
EEGPac06.txt  100      360000
EEGPac07.txt  100      360000
EEGPac08.txt  100      360000
EEGPac09.txt  100      192000

# veeg : 75uv
# Rate : sampling rate in Hz
# Samples : number of samples in file
  
```

Fig. 4.9.- Información relativa a la adquisición de Señales EEG

Para fines del procesamiento se tomó la muestra del archivo "EEGPacO3.txt" (el cual contiene un total de 360,000 muestras de la señal, y a fin de adecuarla y almacenarla en la variable definida que sea capaz de ser leída por el aplicativo Analizador de Espectros, se leyó la información en una etapa intermedia con el aplicativo de excel del MS-Office, y se eliminaron los datos de tiempo que resultan innecesarios, dado que se tiene como dato la frecuencia de muestreo de 100 Hz, así mismo, se tomó solo información del canal 2, dejando de lado la información de los otros tres canales, esto por la limitación del instrumento se ha diseñado para solo analizar una sola señal en el tiempo. La desventaja de usar el excel para esto es que trunca la información hasta los 65536 muestras, que es el número máximo de celdas de este aplicativo del MS Office. Finalmente, la información de la señal fue recogida en el archivo 'EEGPacO3full2ch.txt'.

Ejecución y Resultados de la simulación.-

Ejecutamos el script con los siguientes parámetros:

Número de Muestras	: 31,400
Zoom tiempo	: 100%
Número de puntos FFT	: 1024
Tamaño de ventana análisis	:13,952
Shift	: 0
Ventana MAGX Espectrógrafo	: 100,000
Zoom Frec Espectrografo	: 8,3%
Ventana MAG Espectrograma	: 21,400
Zoom Tiem Espectrograma	: 100%
Zoom Frec Espectrograma:	11%
Archivo de datos fuente	: ' EEGPacO3full2ch.txt '

Con lo cual, tenemos los siguientes resultados:

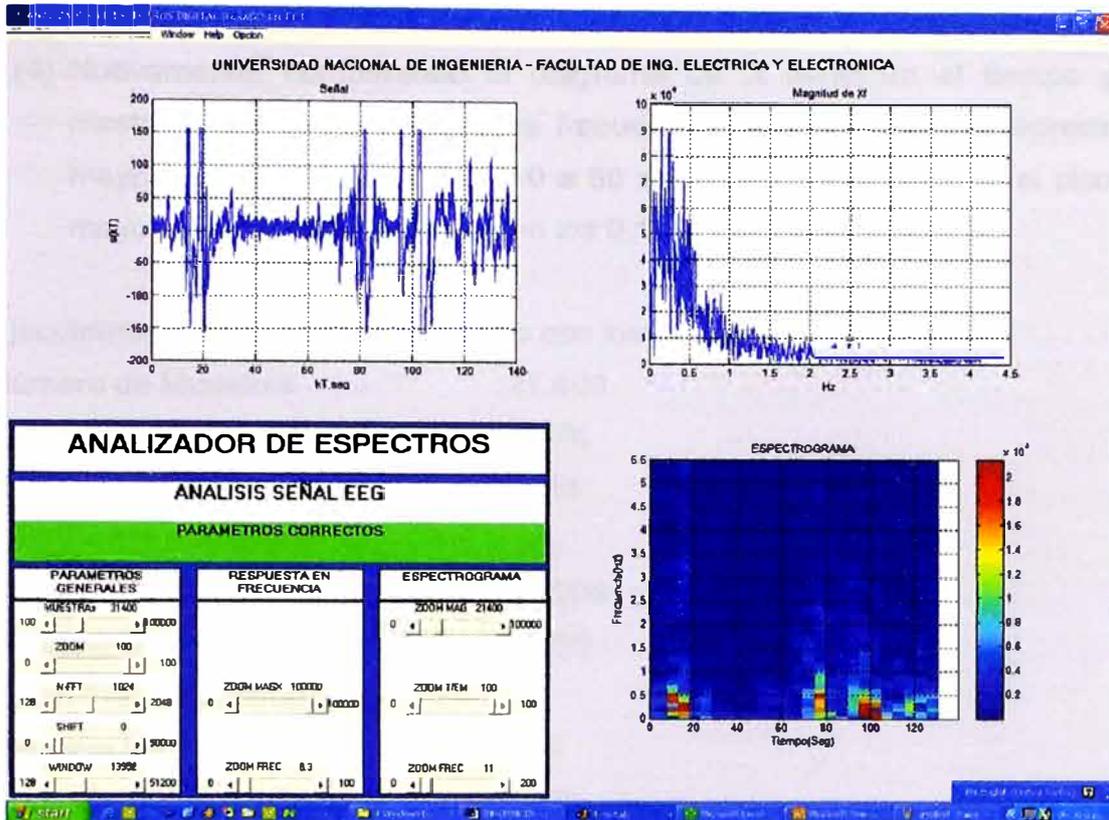


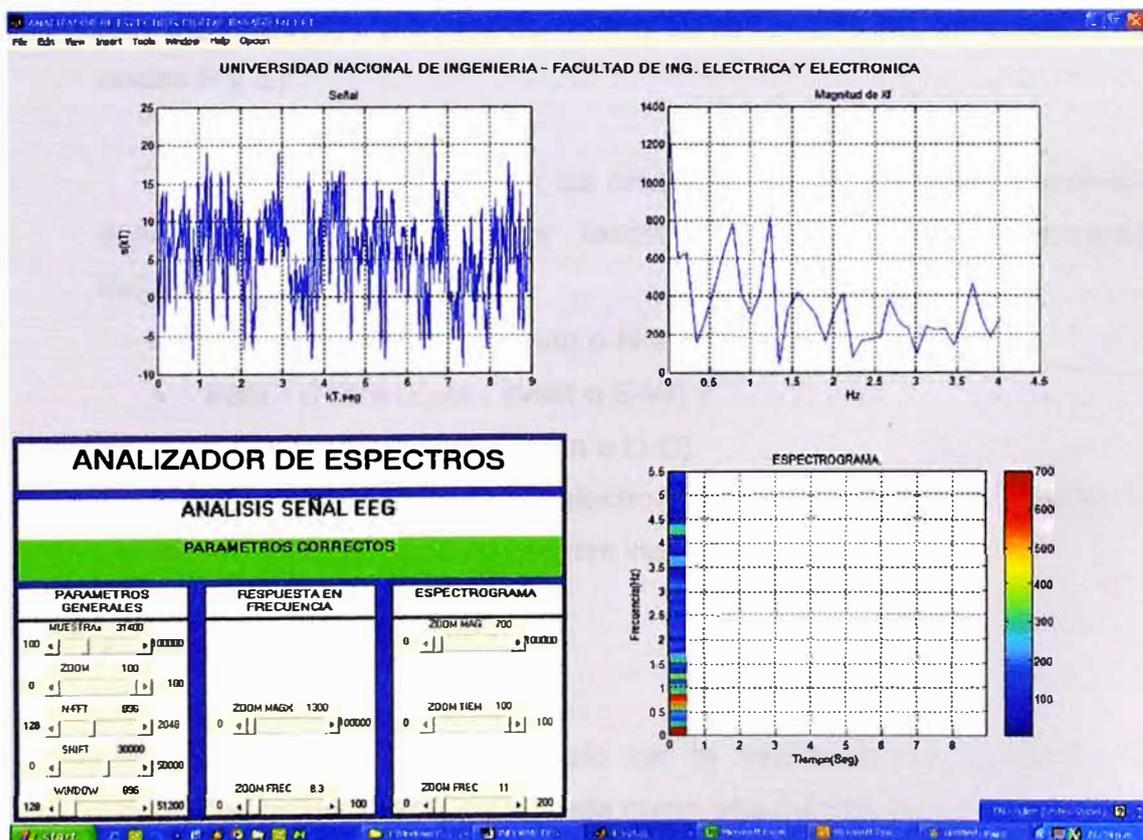
Fig. 4.10.- Resultados del análisis de señal EEG (1)

- (1) El diagrama de la señal ingresada desde el archivo y volcada en la variable de señal S, para un total de 13,952 muestras para la ventana. Se observa que, a diferencia de la señal de electrocardiograma, el componente continuo de se frecuencia cero es prácticamente inexistente, esto porque la señal tiene variable en el tiempo de amplitudes tanto positivas como negativas, por lo que el promedio en el tiempo es aproximadamente cero de amplitud.
- (2) El diagrama de la señal en el dominio de la frecuencia, aplicado el control del zoom del eje de la frecuencia al 8,3% y con un umbral de 68,000 unidades, se aprecia mejor la actividad espectral en la baja frecuencia, siendo las componentes de mayor potencia de 0,1 a 0,4 Hz, teniendo un pico en 0,2 Hz.
- (3) El diagrama de la señal en el Espectrograma, ajustando los parámetros de control indicados, se aprecia la misma actividad espectral ya indicada en el punto anterior, es decir, mayor potencia de la señal en el rango de 0.1 a 0,4 Hz.

(4) Nuevamente, comparando el diagrama de la señal en el tiempo y la mostrada en el espectrograma frecuencia versus tiempo, se aprecia la mayor actividad espectral de 10 a 80 segundos, observándose el pico de mayor potencia precisamente en los 0,15 a 0,22 Hz.

Ejecutamos nuevamente el aplicativo con los siguientes parámetros:

Número de Muestras : 31,400
 Zoom tiempo : 100%
 Número de puntos FFT : 1024
 Tamaño de ventana análisis : 986
 Shift : 30,000
 Ventana MAGX Espectrógrafo : 1,300
 Zoom Frec Espectrograma : 8,3%
 Ventana MAG Espectrograma : 700
 Zoom Tiem Espectrograma : 100%
 Zoom Frec Espectrograma : 11%
 Archivo de datos fuente : ' EEGPacO3full2ch.txt '



Para el nuevo escenario, con 986 muestras de la señal, el procesamiento de la información de la señal y por ende el refresco de la pantalla del instrumento lleva mucho menos tiempo que para un número de muestras más grande.

La señal EEG no tiene la periodicidad mostrada por el ECG, sin embargo, para la pequeña muestra tomada, las frecuencias de más potencia están en un rango de frecuencias de 0,75 y 1,25 Hz.

4.2.3. Señales sismográficas

a. Conceptos sobre las Señales Sismo-gráficas.-

Ondas Sísmicas.-

Las ondas sísmicas, son del tipo de ondas elásticas y son generadas por un impulso tal como el causado por un terremoto o sismo, o por una explosión. Las ondas sísmicas pueden ser propagadas ya sea a lo largo o cerca de la superficie terrestre (Por ejemplo, las ondas de Rayleigh y Love) o a través del interior de la tierra (como por ejemplo, las ondas P y S).

Por su misma naturaleza las ondas sísmicas se pueden manifestar espacialmente; es decir, se tendrán las siguientes componentes tridimensionales:

- Norte - Sur (North - South o N-S),
- Este - Oeste (East - West o E-W) y,
- Arriba - Abajo (Up – Down o U-D).

Por esto mismo, los electrodos que censan la señal se corresponden con las componentes indicadas.

Aceleración.-

Es la velocidad de cambio de la velocidad de un punto de referencia, comúnmente expresada como una fracción o porcentaje de la aceleración de la gravedad ($g=980 \text{ cm/s}^2$).

Acelerograma.-

El registro de un acelerógrafo mostrando la aceleración de la tierra como una función del tiempo.

b. Fuente de Data Adquirida a Analizar.-

La información fuente de naturaleza sísmica, ha sido extraída de la página web del instituto CISMID de la Universidad Nacional de Ingeniería, (12) dentro de su catálogo de señales del acelerograma y adquiridas por las aplicaciones propias del Sismid. Como puede apreciarse en la Tabla 4.1, con los siguientes datos relevantes:

Tabla 4.1.- Información Sismid

DATOS DE LA ESTACIÓN:

NOMBRE DE LA ESTACIÓN : "JORGE ALVA HURTADO"
 CLAVE DE LA ESTACIÓN : CSM
 LOCALIZACIÓN DE LA ESTACIÓN : Universidad Nacional de Ingeniería

DATOS DEL SISMO:

COORDENADAS DE LA ESTACIÓN: 12.01327° LAT. S
 : 77.05021° LONG. W
 ALTITUD(msnm) : 130.00
 TIPO DE SUELO : Grava gruesa
 INSTITUCIÓN RESPONSABLE : CISMID-UNI
 FECHA DEL SISMO : 29 de julio de 2004
 HORA EPICENTRO (Local) : 5:00:00
 COORDENADAS DEL EPICENTRO : 11.19°LAT. S, : 78.24°LONG. W
 PROF. FOCAL /DIST. EPIC. /DIST. HIPOC. (Km) : 61.00 160.60 171.79
 MAGNITUD : ML mb MS MW
 : 5.4*

DATOS DEL ACELERÓGRAFO:

MODELO DEL ACELERÓGRAFO : Acelerógrafo Digital Kinemetrix-Modelo ETNA
 NÚMERO DE SERIE DEL ACELERÓGRAFO :
 NÚMERO DE CANALES : 3
 ORIENTACIÓN, Canal1/Canal2/Canal3 : E-W N-S U-D
 FRECUENCIA DE MUESTREO, Canal1/Canal2/Canal3 (muestras/s): 200 200 200
 MÁXIMA AMPLITUD DE SENSORES, Canal1/Canal2/Canal3 (g) : 2.00 2.00 2.00
 PERÍODO DEL INSTRUMENTO, Canal1/Canal2/Canal3 (s) : 0.0049 0.0046 0.005
 AMORTIGUAMIENTO DE SENSORES, Canal1/Canal2/Canal3 : 0.70 0.70 0.70
 UMBRAL DE DISPARO, Canal1/Canal2/Canal3 (Gal) : 2.00 2.00 2.00
 MEMORIA DE PRE-EVENTO (s) : 15 15 15
 TIEMPO DE POST-EVENTO (s) : 20 20 20

Donde se puede observar que la frecuencia de muestreo es de 200 Hz.

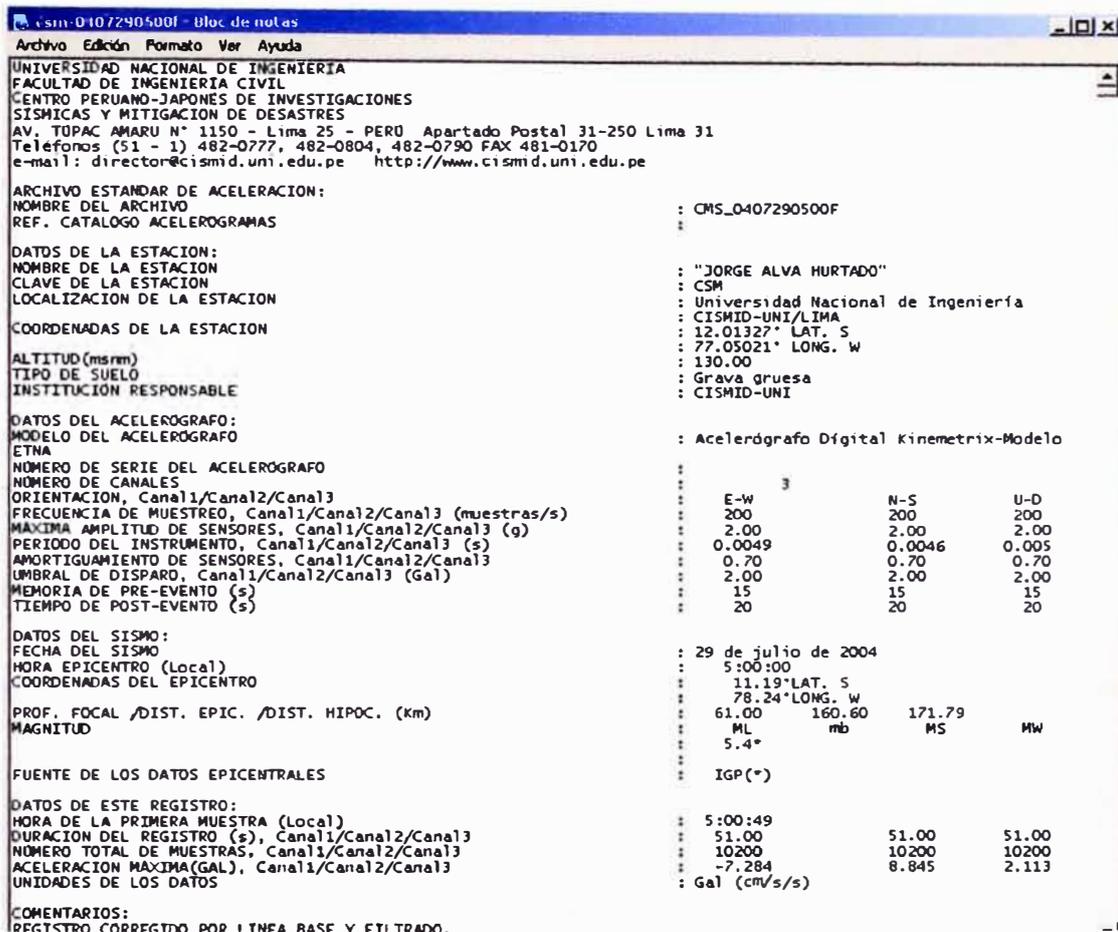


Fig. 4.13.- Información relativa a la adquisición de Señales Sísmicas

Para fines del procesamiento se tomó la muestra del archivo moq1-0106231533f-EW.txt (la cual contiene un total de 10,200 muestras de la señal y la frecuencia de muestreo de 200 Hz), y a fin de adecuarla y almacenarla en la variable definida que sea capaz de ser leída por el aplicativo Analizador de Espectros, se leyó la información en una etapa intermedia con el aplicativo de excel, y se eliminaron los datos que resultan innecesarios, dejando solo información del canal 1 que contiene la información E-W (Este-Oeste), desechando las demás, puesto que, como lo ya indicado, el instrumento solo es capaz de procesar señales de un solo canal. La información de la señal fue recogida en el archivo "moq1-0106231533f-EW.txt".

Ejecución y Resultados de la simulación.-

Ejecutamos el script con los siguientes parámetros:

Número de Muestras	: 31,600
Zoom tiempo	: 100%
Número de puntos FFT	: 1024
Tamaño de ventana análisis	: 16,256
Shift	: 0
Ventana MAGX Espectrógrafo	: 53,000
Zoom Frec Espectrograma	: 7,7%
Ventana MAG Espectrograma	: 11,300
Zoom Tiem Espectrograma	: 100%
Zoom Frec Espectrograma:	11%
Archivo de datos fuente	: ' moq1-0106231533f-EW.txt'

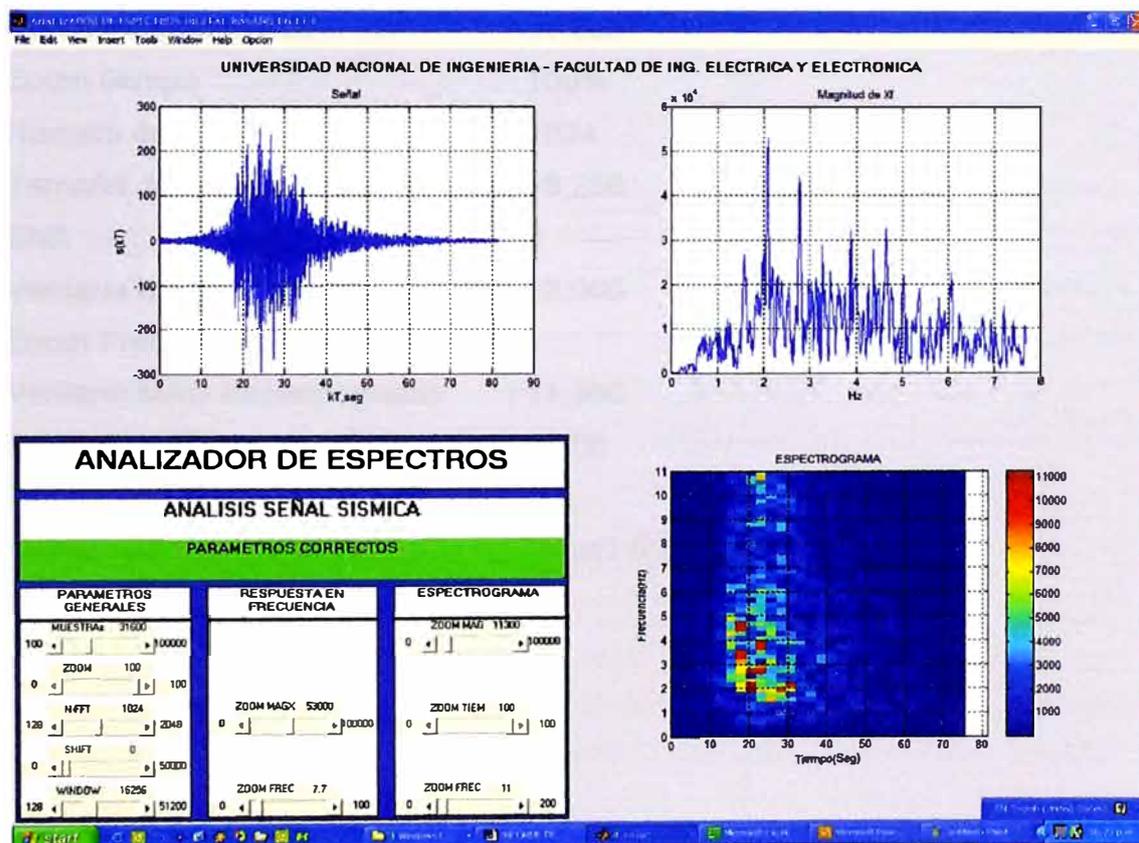


Fig. 4.14.- Pantalla con resultados SIM para datos mostrados (1)

- (1) El diagrama de la señal ingresada desde el archivo y volcada en la variable de señal S, para un total de 16,256 muestras.

- (2) El diagrama de la señal en el dominio de la frecuencia, aplicado el control del zoom del eje de la frecuencia al 7,7%, desechando los componentes alrededor del 100 Hz, se verifica la alta actividad espectrográfica para frecuencias entre 2,1, 2,8, 3,8 y 4,7 Hz, específicamente, el pico más alto está alrededor de 2,1 Hz.
- (3) El diagrama de la señal en el Espectrograma, para las condiciones de parámetros elegida, se aprecia igualmente una mayor intensidad de potencia de la señal alrededor de los 2 a 5 Hz, como ya lo habíamos comprobado en el espectrógrafo, pero también se aprecia potencia significativa a 10,7 Hz. De aquí también se observa que la duración del movimiento telúrico ha sido de aproximadamente 20 segundos.

Ejecutamos nuevamente el script con los siguientes parámetros:

Número de Muestras	: 31,600
Zoom tiempo	: 100%
Número de puntos FFT	: 1024
Tamaño de ventana análisis	: 16,256
Shift	: 0
Ventana MAGX Espectrógrafo	: 53,000
Zoom Frec Espectrografo	: 7,7%
Ventana MAG Espectrograma	: 11,300
Zoom Tiem Espectrograma	: 100%
Zoom Frec Espectrograma:	11%
Archivo de datos fuente	: ' moq1-0106231533f-EW.txt'

CONCLUSIONES Y APORTES

1. Analizador de espectros – Modulo de calibración

En todo proceso de diseño e implementación de sistemas de medición o análisis en general, es necesaria la inclusión de algún módulo de simulación o calibración, que permita que, a través de señales patrón con sus características principales plenamente conocidas, verificar la correcta respuesta de dicho sistema y comprobar que sus salidas se correspondan con lo esperado teóricamente.

Para el módulo de calibración presentado, se puede verificar la correcta respuesta a las señales patrón ingresadas; tanto en la simple graficación de la misma, donde se puede apreciar el comportamiento de la señal en el tiempo; así como la respuesta en frecuencia de la misma, incluso verificándose las componentes de potencia de señal respecto del tiempo de la frecuencia, en la pantalla del espectrograma. Dada la señal senoidal uniforme, y para la cantidad de muestras tomadas, las frecuencias componentes de la señal son puras, y eso se verifica en la pantalla de analizador de frecuencias. Así mismo, en la pantalla del espectrograma, se puede apreciar las frecuencias que se mantienen constantes en el tiempo.

Respecto al mismo instrumento, se puede verificar que, para una mejor precisión en el análisis de las señales, se requiere un mayor número de muestras de la señal, y un mayor número de (N) de puntos de evaluación de la Transformada Rápida de Fourier. Lo cual se comprueba de los dos grupos de resultados mostrados.

Sin embargo, como ya se había advertido, debe haber un compromiso entre la precisión y la respuesta en tiempo real y debe establecerse límites adecuados, puesto que, un número muy elevado de muestras, podría llevar al uso de demasiada memoria y mayor tiempo en el procesamiento de la señal; así mismo, para un número

muy elevado de puntos de evaluación del FFT, se requeriría un mayor tiempo de procesamiento, y la respuesta podría no ser la esperada. En el caso particular de la aplicación, no se apreciaría mucho la lentitud en el procesamiento, pues se está analizando señales que ya han sido adquiridas y almacenadas en archivo, sobre las que se aplica el procesamiento y presentan resultados en forma gráfica como resultado de dicho procesamiento, pero el efecto sería sumamente notorio si se aplicara a señales que adquiridas en tiempo real y que se desean analizar espectralmente también en tiempo real.

2. Analizador de espectros – Análisis de ECG

Verificadas las correctas respuestas del módulo de calibración, la primera aplicación utilizada es la del módulo ECG. Se puede apreciar que, para las condiciones indicadas, la señal ECG es sumamente compleja en cuanto a sus componentes espectrales, y dado el alto componente en frecuencia cero (el promedio de la señal está alrededor de 150 unidades, como se puede apreciar en el diagrama de la respuesta de la señal respecto al tiempo), esta aparece con una amplitud alta, tanto en la pantalla del analizador de espectros, como en el espectrograma, y no permite apreciar claramente las otras componentes de frecuencia.

Aquí la importancia de los módulos del sistema, que permiten hacer acercamiento (zoom) dentro de los gráficos de respuesta en frecuencia (Del analizador de espectros y el espectrograma), con lo cual se puede apreciar que las componentes de frecuencia de mayor potencia están debajo de 1 Hz, para ser más específicos, el pico de la frecuencia de mayor potencia (después de la componente de frecuencia cero), está alrededor de los 0,3 Hz, según se puede apreciar en los diagramas indicados.

En la pantalla del espectrograma, se puede apreciar las tonalidades más intensas en el tiempo, según los cambios que se producen en el comportamiento de la señal en el tiempo. La presentación de mosaicos del mapa de colores se aprecian en forma gruesa, lo cual es aún más apreciable cuando se hace el acercamiento en el dominio de la frecuencia y en el tiempo, lo cual puede ser corregido con la inclusión de mayor número de puntos de evaluación del FFT, según se puede verificar comparando los dos grupos de resultados.

3. Analizador de espectros – Análisis de EEG

Del análisis de una señal de electroencefalograma también se concluye que tiene un espectro con componentes de mayor potencia en muy bajas frecuencias, y a diferencia de la señal de electrocardiograma y de la señal sísmica, sus amplitudes son alternas de positivo a negativo, por lo que la componente para frecuencia cero es prácticamente nula, lo cual puede apreciarse en cada una de las pantallas del instrumento, es decir, las que muestran el diagrama de señal en el tiempo, el diagrama de la señal en la frecuencia y el espectrograma.

Se concluye que la actividad espectral de las señales de electroencefalogramas tiene componentes de mayor potencia en los rangos de 0,1 a 0,4 Hz.

El diagrama de la señal en el Espectrograma, ajustando los parámetros de control indicados, se aprecia la misma actividad espectral ya indicada en el punto anterior, es decir, mayor potencia de la señal en el rango de 0.1 a 0,4 Hz.

Adicionalmente, las señales electroencefalográficas tienen patrones repetitivos en el tiempo, aunque no a períodos constantes, que según lo analizado en el caso, puede tener períodos de 20 a 80 segundos.

4. Analizador de espectros – Análisis de señales sísmicas

Para el caso particular del análisis espectral para la naturaleza de la señal sísmica, recibida del acelerómetro, la respuesta del sistema es adecuada.

Del diagrama de la señal en función del tiempo, se concluye que al ser la sísmica, es oscilante alrededor de cero. Se aprecia el inicio de la actividad sísmica a los 15 segundos y finaliza a los 35 segundos.

Así mismo, se verifica que hay una actividad espectrográfica importante en la baja frecuencia, como era de esperarse, y está es más intensa de 2,1 a 4,7 Hz, y una ligera actividad a los 10.7 Hz. Se tomó otra muestra del acelerómetro de Moquegua (movimiento sísmico del 23 de Junio del 2001).

Utilizando las bondades del instrumento digital, se ha podido analizar el comportamiento de otra señal de baja frecuencia, y extraer información en cuanto a su naturaleza. Para el contexto dado, los resultados del análisis resultan valiosos, pues permite, entre otros tantos usos, utilizarla para probar materiales de construcción, que puedan ser resistentes cuando se los someta a vibraciones a las frecuencias típicas que produce un movimiento sísmico.

5. Analizador de espectros – Conclusiones generales

Para cada señal en particular, es preciso tomar en cuenta la frecuencia de señal mínima que debe ser considerada en la muestra, en otras palabras, es requerido indicar el tamaño de muestra mínimo, que contenga la información completa de la señal en particular.

Así, por ejemplo, en la señal ECG, lo mínimo a registrar y muestrear sería un ciclo completo de señal, es decir, que al menos contenga un intervalo RR, que es algo menos de 1 segundo, pero lo ideal sería tomar muestras por el espacio de 10 segundos.

Por otro lado, el analizar un tamaño de muestra muy largo, también podría ser un desperdicio, dado que puede perderse el significado de la señal en si. Esto se explica porque, para una muestra muy grande de señal, pueden presentarse muchos ruidos o señales espurias, que introduzcan información incorrecta. Por ello la efectividad del instrumento diseñado, pues permite a través de la ventana deslizante, escoger una porción importante y significativa de la señal y analizarla.

ANEXO A

TEOREMA DE PARSEVAL

En matemáticas, el teorema de Parseval usualmente se refiere al resultado que la transformada de Fourier es unitaria; simplemente, que la suma (o la integral) del cuadrado de una función es igual a la suma (o la integral) del cuadrado de su transformada. Esto se origina desde un teorema acerca de las series formulado por Marc-Antonie Parseval en 1799, el cual más tarde fue aplicada a las series de Fourier.

A pesar del término "Teorema de Parseval" es a menudo aplicado para describir la característica unitaria de cualquier transformada de Fourier, especialmente en física e ingeniería, la forma más general de esta propiedad es más propiamente llamada Teorema de Plancherel.

Teorema

El teorema original, llevado a la forma moderna, puede ser formulado como sigue:

Suponiendo que se tienen las dos series:

$$A(z) = \sum_{n=-\infty}^{\infty} a_n z^n$$

$$B(z) = \sum_{n=-\infty}^{\infty} b_n z^n$$

En algunos coeficientes a_n y b_n (tomados como coeficientes complejos, a pesar que Parseval solo considera coeficiente real empezando en $n=0$, y despreciando la condición de cuando la serie converge). El teorema establece que:

$$\sum_{n=-\infty}^{\infty} a_n b_n^* = \frac{1}{2\pi} \int_{-\pi}^{\pi} A(e^{i\phi}) B(e^{i\phi})^* d\phi$$

Cuando i es la unidad imaginaria y $*$ denota la conjugación compleja. Parseval en realidad presenta el teorema sin una prueba, considerando esta como evidente por sí misma.

Existen varios casos especiales importantes del teorema. (1) Primero, para A y B de la misma serie, se obtiene que:

$$\sum_{n=-\infty}^{\infty} |a_n|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |A(e^{i\phi})|^2 d\phi$$

del cual la sigue la unitariedad de la serie de Fourier, donde a_n corresponde al coeficiente F_n de ña serie de Fourier de la función $f(x) = A(e^{ix})$.

En particular, uno a menudo considera solo la serie de Fourier para funciones de valor real (o para A real y B para todo ϕ), el cual corresponde al caso especial: a_0 real,

$a_{-n} = a_n^*$, b_0 real, y $b_{-n} = b_n^*$. En este caso:

$$a_0 b_0 + 2\Re \sum_{n=1}^{\infty} a_n b_n^* = \frac{1}{2\pi} \int_{-\pi}^{\pi} A(e^{i\phi}) B(e^{i\phi}) d\phi$$

Donde \Re denota la parte real. (En la notación de la Serie de Fourier, se reemplaza a_n y b_n , por $a_n/2 - ib_n/2$).

ANEXO B

TEORÍA DEL MUESTREO

Introducción:

El muestreo es un método fundamental para todo procesamiento de señal digital indistintamente de sus aplicaciones, y se puede definir como un proceso de medición de una señal analógica en distintos puntos (variables discretas de tiempo).

La representación digital de señales analógicas ofrece una serie de ventajas en términos de:

- Inmunidad frente a los efectos del ruido, por lo cual, tratándose de señales digitales, esto significa que se puede enviar muchos más bps.
- Uso de equipo de procesamiento flexible, en particular la computadora o cualquier procesador digital especializado.
- Mayor confiabilidad en los equipos de procesamiento.
- Se hace más fácil de adaptar algoritmos complejos.

La Teoría del Muestreo:

La teoría del muestreo tuvo sus fundamentos en las teorías de Claude E. Shannon y Harry Nyquist. El primero, llegó a ser llamado el padre de la teoría de la información, principalmente por su trabajo sobre "Teoría de las Matemáticas para las Comunicaciones". Nyquist fue el primero en establecer la teoría del muestreo en 1928, pero no fue probada sino hasta que Shannon lo fundamentó en su trabajo "Comunicaciones en la presencia del ruido", 21 años más tarde.

Definición:

"Cuando se muestrea una señal analógica la frecuencia de muestreo debe ser mayor que dos veces la componente de frecuencia más alta de la señal analógica, para que se sea capaz de reconstruir la señal original desde la versión muestreada de la señal".

Notación:

Las siguientes notaciones suelen ser usadas referidas a la teoría del muestreo de señales.

- Señal analógica original: $x(t)$.
- Frecuencia de Muestreo: F_s
- Intervalo de Muestreo: T_s

Donde:

$$F_s = \frac{1}{T_s}$$

- Señal muestreada $x_s(n)$. (Notar que: $x_s(n) = x(nT_s)$)
- La frecuencia angular real Ω .
- La frecuencia angular digital ω . (Donde: $\omega = \Omega T_s$)

B.1 Ejemplo en MathLab de la teoría del Muestreo

```
% Muestreo y reconstrucción
% Creando la Señal Analógica
```

```
clear; %Borrar todas las variables
```

```
t=0:.1:20;
```

```
F1=.1;
```

```
F2=.2;
```

```
x=sin(2*pi*F1*t)+sin(2*pi*F2*t);
```

```
%Graficación
```

```
figure(1);
```

```
subplot(2,1,1);
```

```
plot(t,x);
```

```
title('Original signal')
```

```
xlabel('t');
```

```
ylabel('x(t)');
```

```
subplot(2,1,2);
```

```

x_samples=x(1:10:201); %gets 21 samples of x.
stem(x_samples,'filled');
title('Sampled signal')
xlabel('n');
ylabel('x_s(n)');
axis([0 20 -2 2]);

%creating dialog box with explanations
l1=[blanks(10),'Sample by sample reconstruction.'];
l2='Blue dots: Input samples.';
l3='Blue curve: reconstructed signal.';
l4='Red curve: contribution to output sample from current sample.';
l5='Press any key to update with 1 iteration.';
l6='(You can keep this window open while watching the reconstruction)';
information = {l1,"",l2,l3,l4,"",l5,"",l6};

%starting reconstruction process
figure(2);
messagebox=msgbox(information,'Information','help');
subplot(2,1,2);
plot(t,x,'black');
hold on;
plot([0 20],[0 0],'black');
hold off;
xlabel('t');
ylabel('x(t)');
title('Original signal');
grid;

x_recon=0;
subplot(2,1,1);
for k=0:length(x_samples)-1
    stem(0:length(x_samples)-1,x_samples,'filled');
    if k==length(x_samples)-1
        title('Reconstruction finished');
    else

```

```

    title('Sample by sample reconstruction');
end
grid on;
l=k:-.1:-20+k;
x_recon=x_recon+x_samples(k+1)*sinc(l);
axis([0 20 -2 2]);
hold;
plot(t,x_samples(k+1)*sinc(l),'r')
plot(t,x_recon);
hold off;
waitforbuttonpress;
end

```

Resultados:

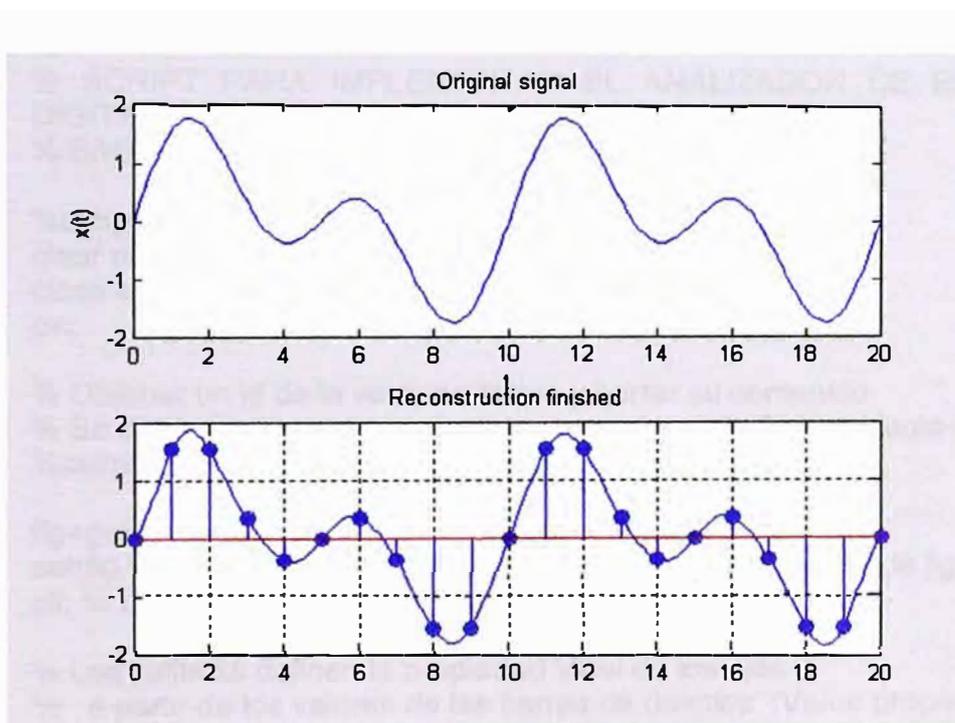


Figura B.1 Señal analógica original y la señal muestreada

ANEXO C

DESARROLLO DE SCRIPTS EN MATLAB

C.1. Implementación de algoritmos – Programa de calibración

C.1.1. Módulo Interactivo con PC.-

Script:

```
%*****  
% SCRIPT PARA IMPLEMENTAR EL ANALIZADOR DE ESPECTROS  
DIGITAL  
% BASADO EN LA APLICACION DE LA FFT  
  
%Limpiado de variables y parametros  
clear all;  
close all;  
clc;  
  
% Obtener un id de la ventana activa y borrar su contenido  
% Se define solo una ventana, la que proporcionara el ambiente interactivo y  
%asimismo mostrara la señal y los resultados.  
  
fig=gcf;  
set(fig,'Position',[6 36 1260 940]); % Se define el tamaño de la figura 1  
clf; % Limpia contenido de la figura 1  
  
% Los callbcks definen la propiedad View de los ejes  
% a partir de los valores de las barras de desplaz. (Value property)  
% y escriben sus valores en los controles texto  
  
%Definicion de inicial de parametros claves  
f=200;  
Fs=1024;  
res=1;  
N=1024;  
NN=10; %Definicion del numero de ondas de la mas baja frecuencia  
ts=1/f; %periodo de la señal f  
tmax=max(ts);
```

```

t=0:1/Fs:(tmax*NN);
Ns=Fs*(tmax*NN); % El numero de muestras total;
Ts=1/Fs ; % periodo de señal de muestreo, o periodo entre muestreos

% DELIMITACION DEL CUADRO DE CONTROL

ft_dir1=icontrol(fig,...
    'Style','frame',...
    'BackgroundColor','White',...
    'Position',[9 9 192 242]);

% DEFINICION DE LA BARRA QUE CONTROLA LA RESOLUCION

sli_a1=icontrol(fig,'Style','slider','Position',[40           10           120
20],'Min',0,'Max',100,'Value',100,...

'Callback','set(a1_cur,"String",num2str(get(sli_a1,"Val"))),res=get(sli_a1,"Val'
')/100;Scriptespectrovar');

% Definir los controles de texto para los valores minimos
a1_min=icontrol(fig,...
    'Style','text',...
    'Pos',[10 10 30 20],...
    'String',num2str(get(sli_a1,'Min')));

% Definir los controles de texto para los valores maximos
a1_max=icontrol(fig,...
    'Style','text',...
    'Pos',[160 10 30 20],...
    'String',num2str(get(sli_a1,'Max')),...
    'Horiz','right');

% Definir las etiquetas de las barras
a1_label=icontrol(fig,...
    'Style','text',...
    'Pos',[40 30 69 20],...
    'String','RESOL');

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
a1_cur=icontrol(fig,...
    'Style','text',...
    'Pos',[110 30 50 20],...
    'String',num2str(get(sli_a1,'Value')));

% DEFINICION DE LA BARRA QUE CONTROLA LA FRECUENCIA DE
MUESTREO

sli_Fs=icontrol(fig,'Style','slider','Position',[40           60           120
20],'Min',0,'Max',100000,'Value',1024,...

```

```
'Callback','set(Fs_cur,"String",num2str(get(sli_Fs,"Val"))),Fs=get(sli_Fs,"Val")
);Scriptespectrovar');
```

```
% Definir los controles de texto para los valores minimos
```

```
Fs_min=icontrol(fig,...
    'Style','text',...
    'Pos',[10 60 30 20],...
    'String',num2str(get(sli_Fs,'Min')));
```

```
% Definir los controles de texto para los valores maximos
```

```
Fs_max=icontrol(fig,...
    'Style','text',...
    'Pos',[160 60 30 20],...
    'String',num2str(get(sli_Fs,'Max')),...
    'Horiz','right');
```

```
% Definir las etiquetas de las barras
```

```
Fs_label=icontrol(fig,...
    'Style','text',...
    'Pos',[40 80 69 20],...
    'String','Frec Muestreo');
```

```
% Definir los controles de texto para los valores actuales
```

```
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
```

```
Fs_cur=icontrol(fig,...
    'Style','text',...
    'Pos',[110 80 50 20],...
    'String',num2str(get(sli_Fs,'Value')));
```

```
% DEFINICION DE LA BARRA QUE CONTROLA EL NUMERO DE
PUNTOS DE LA FFT
```

```
sli_N=icontrol(fig,'Style','slider','Position',[40          110          120
20],'Min',0,'Max',2048,'Value',1024,...
```

```
'Callback','set(N_cur,"String",num2str(get(sli_N,"Val"))),N=get(sli_N,"Val");S
criptespectrovar');
```

```
% Definir los controles de texto para los valores minimos
```

```
N_min=icontrol(fig,...
    'Style','text',...
    'Pos',[10 110 30 20],...
    'String',num2str(get(sli_N,'Min')));
```

```
% Definir los controles de texto para los valores maximos
```

```
N_max=icontrol(fig,...
    'Style','text',...
    'Pos',[160 110 30 20],...
    'String',num2str(get(sli_N,'Max')),...
    'Horiz','right');
```

```

% Definir las etiquetas de las barras
N_label=icontrol(fig,...
    'Style', 'text',...
    'Pos',[40 130 69 20],...
    'String','NPto FFT');

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
N_cur=icontrol(fig,...
    'Style', 'text',...
    'Pos',[110 130 50 20],...
    'String',num2str(get(sli_N,'Value')));

% DEFINICION DE LA BARRA QUE CONTROLA LA FRECUENCIA DE
SEÑAL

sli_f=icontrol(fig,'Style','slider','Position',[40                160                120
20],'Min',0,'Max',10000,'Value',200,...

'CallBack','set(f_cur,"String",num2str(get(sli_f,"Val"))),f=get(sli_f,"Val");Script
espectrovar');

% Definir los controles de texto para los valores minimos
f_min=icontrol(fig,...
    'Style','text',...
    'Pos',[10 160 30 20],...
    'String',num2str(get(sli_f,'Min')));

% Definir los controles de texto para los valores maximos
f_max=icontrol(fig,...
    'Style','text',...
    'Pos',[160 160 30 20],...
    'String',num2str(get(sli_f,'Max')),...
    'Horiz','right');

% Definir las etiquetas de las barras
f_label=icontrol(fig,...
    'Style', 'text',...
    'Pos',[40 180 69 20],...
    'String','Frec Senal');

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
f_cur=icontrol(fig,...
    'Style', 'text',...
    'Pos',[110 180 50 20],...
    'String',num2str(get(sli_f,'Value')));

```

```
% DEFINICION DE LA BARRA QUE CONTROLA EN NUMERO DE PERIODOS
```

```
sli_NN=icontrol(fig,'Style','slider','Position',[40 210 20]...'Min',0,'Max',100,'Value',10,...
```

```
'CallBack','set(NN_cur,"String",num2str(get(sli_NN,"Value"))),NN=get(sli_NN,"Value");Scriptespectrovar');
```

```
% Definir los controles de texto para los valores minimos
```

```
NN_min=icontrol(fig,...
    'Style','text',...
    'Pos',[10 210 30 20],...
    'String',num2str(get(sli_NN,'Min')));
```

```
% Definir los controles de texto para los valores maximos
```

```
NN_max=icontrol(fig,...
    'Style','text',...
    'Pos',[160 210 30 20],...
    'String',num2str(get(sli_NN,'Max')),...
    'Horiz','right');
```

```
% Definir las etiquetas de las barras
```

```
NN_label=icontrol(fig,...
    'Style','text',...
    'Pos',[40 230 69 20],...
    'String','Num Per');
```

```
% Definir los controles de texto para los valores actuales
```

```
% Los valores son inicializados aqui y son modificados por los callbacks de las barras cuando el usuario cambia sus valores
```

```
NN_cur=icontrol(fig,...
    'Style','text',...
    'Pos',[110 230 50 20],...
    'String',num2str(get(sli_NN,'Value')));
```

C.1.2. Módulo Espectrógrafo (Espectro FFT).-

```
% *****
%***** ANALIZADOR DE ESPECTROS DIGITAL *****
% Scriptespectrovar.m
% *****
```

```
a=1; % Define la amplitud de la señal
ts=1/f; %periodo de la señal base
tmax=max(ts);
```

```
%while Fs < 2*f,
```

```

% disp('Frecuencia incorrecta, considerar condicion de Nyquist');
% Fs=input('Ingrese la Frecuencia de Muestreo=');
%end;

t=0:1/Fs:(tmax*NN);
Ns=Fs*(tmax*NN); % El numero de muestras total;
Ts=1/Fs ; % periodo de señal de muestreo, o periodo entre muestreos

s=a*cos(2*pi*f*t);

subplot(2,2,1),
plot(t,s),title('Señal');
xlabel('kT,seg');
ylabel('s(kT)');
grid;

% *****
% Redondeo de parametros seleccionados *
% *****

f=round(f);
Fs=round(Fs);
N=round(N);
NN=round(NN); %Definicion del numero de ondas de la mas baja frecuencia

ts=1/f; %periodo de la señal f
tmax=max(ts);
k=0:1:Ns-1;

% *****
% Usando la FTT simple *
% *****

F=fft(s);
Magx=abs(F);
hertz=k*(1/(Ns*Ts));

%subplot(2,2,2),stem(hertz(1:round(Ns*res)),Magx(1:
round(Ns*res))),title('Magnitud de Xf'),
%xlabel('Hz'),
%grid;
subplot(2,2,2),plot(hertz(1:round(Ns*res)),Magx(1:
round(Ns*res))),title('Magnitud de Xf'),
xlabel('Hz'),
grid;

% *****
% ***** ESPECTROGRAMA *****
% *****

ww=hamming(N);
subplot(2,2,4),
[B,Fre,Tiem]=specgram(s,N,Fs,ww);
imagesc(Tiem,Fre,abs(B)), axis xy;
axis ([0 max(t) 0 max(Fre)]), colormap(jet);

```

```

colorbar('vert');
xlabel('Tiempo(Seg)'),ylabel('Frecuencia(Hz)');
abcd = sprintf('ESPECTROGRAMA');
set(get(gca,'title'),'String',abcd);
zoom on;
grid;

```

C.2. Implementación de algoritmos – Programa de Aplicación

C.2.1. Módulo Interactivo con PC.-

Script:

```

%*****
% SCRIPT PARA IMPLEMENTAR EL ANALIZADOR DE ESPECTROS
DIGITAL
% BASADO EN LA APLICACION DE LA FFT
%*****
% MODULO DE CONTROL DE INTERFAZ GRAFICA USUARIO
% AUTOR: OSCAR TORRES MEJIA
% FECHA: FEBRERO 2005
% FILE : ANALESPECG.m
% V CURSO DE TITULACION FIEE-UNI
%*****

%Limpiado general de variables y parametros
clear all;

close all;

clc; % pantalla del Command window

%Obtener un id de la ventana activa y borrar su contenido
fig=gcf;
set(fig,'Position',[6 36 1260 940]); % Se define el tamaño de la figura 1

```

```

set(fig,'Name','ANALIZADOR DE ESPECTROS DIGITAL BASADO EN
FFT'); % Se define el tamaño de la figura 1
set(fig,'Numbertitle','off');
%set(fig,'menubar','none');
clf % Limpia pantalla de la aplicacion

% DEFINICION DE MENUES

% Menu Opcion
id_Opcion=uimenu(fig,'Label','&Opcion',...
    'Accelerator','o');

% Creacion de los diferentes submenues

id_ECG=uimenu(id_Opcion,'Label','ECG',...
    'Callback','opcion=1;');

id_EEG=uimenu(id_Opcion,'Label','EEG',...
    'Callback','opcion=2;');

id_SISM=uimenu(id_Opcion,'Label','SISM',...
    'Callback','opcion=3;');

id_PATRON=uimenu(id_Opcion,'Label','PATRON',...
    'Callback','opcion=4;');

%DEFINICION INICIAL DE PARAMETROS CLAVE

N=1024; % Numero de puntos de evaluacion FFT
Ns=2000; %Numero de muestras
zoomf=1; % zoom de espectrometro al 100%.
zoomt=1; % Zoom de respuesta de la señal en el tiempo al 100%

```

```
Ws=1024; % Tamaño de ventana por default es igual al numero de
muestras total
```

```
Shift=0; % Inicia con desplazamiento de la ventana en cero
```

```
zoomFre=1;
```

```
zoomST=1;
```

```
umbral_XF=100000;
```

```
umbral_B=100000;
```

```
% DELIMITACION DE CUADRO DE CONTROL
```

```
ft_dir0=uicontrol(fig,...
```

```
    'Style','frame',...
```

```
    'BackgroundColor','blue',...
```

```
    'Position',[5 5 620 471]);
```

```
ft_dir1=uicontrol(fig,...
```

```
    'Style','frame',...
```

```
    'BackgroundColor','White',...
```

```
    'Position',[9 9 192 242]);
```

```
ft_dir1t=uicontrol(fig,...
```

```
    'Style','frame',...
```

```
    'BackgroundColor','White',...
```

```
    'Position',[9 252 192 40]);
```

```
ft_dir1_label=uicontrol(fig,...
```

```
    'Style','text',...
```

```
    'BackgroundColor','White',...
```

```
        'FontSize',10,...
```

```
    'FontWeight','bold',...
```

```
        'HorizontalAlignment','center',...
```

```
    'Position',[10 253 190 38],...
```

```
    'String','PARAMETROS GENERALES');
```

```
ft_dir2=uicontrol(fig,...
    'Style','frame',...
    'BackgroundColor','White',...
    'Position',[219 9 192 242]);

ft_dir2t=uicontrol(fig,...
    'Style','frame',...
    'BackgroundColor','White',...
    'Position',[219 252 192 40]);

ft_dir2_label=uicontrol(fig,...
    'Style','text',...
    'BackgroundColor','White',...
    'FontSize',10,...
    'FontWeight','bold',...
    'HorizontalAlignment','center',...
    'Position',[220 253 190 38],...
    'String','RESPUESTA EN FRECUENCIA');

ft_dir3=uicontrol(fig,...
    'Style','frame',...
    'BackgroundColor','White',...
    'Position',[429 9 193 242]);

ft_dir3t=uicontrol(fig,...
    'Style','frame',...
    'BackgroundColor','White',...
    'Position',[429 252 193 40]);

ft_dir3_label=uicontrol(fig,...
    'Style','text',...
    'BackgroundColor','White',...
    'FontSize',10,...
    'FontWeight','bold',...
    'HorizontalAlignment','center',...
    'Position',[430 253 190 38],...
```

```
'String','ESPECTROGRAMA');
```

```
ft_dir4=uicontrol(fig,...
    'Style','frame',...
    'BackgroundColor','White',...
    'Position',[9 300 613 100]);
```

```
ft_dir5=uicontrol(fig,...
    'Style','frame',...
    'FontName','Arial',...
    'FontSize',20,...
    'BackgroundColor','White',...
    'Position',[9 410 613 60]);
```

```
ft_dir5_label=uicontrol(fig,...
    'Style','text',...
    'BackgroundColor','White',...
    'FontSize',24,...
    'FontWeight','bold',...
    'HorizontalAlignment','center',...
    'Pos',[10 415 610 50],...
    'String','ANALIZADOR DE ESPECTROS');
```

```
ft_dir6_label=uicontrol(fig,...
    'Style','text',...
    'BackgroundColor',[0.8 0.8 0.8],...
    'FontSize',12,...
    'FontWeight','bold',...
    'HorizontalAlignment','center',...
    'Position',[10 920 1240 16],...
    'String','UNIVERSIDAD NACIONAL DE INGENIERIA - FACULTAD DE
    ING. ELECTRICA Y ELECTRONICA');
```

```
% SCRIPT PARA DEFINIR LOS VALORES Y PARAMETROS
SELECCIONABLES A TRAVES DE
% BARRA DE DESPLAZAMIENTOS EN LA INTERFAZ GRAFICA
```

```
% I.- CONTROLES GENERALES
```

```
% A. DEFINICION DE LA BARRA QUE CONTROLA EL NUMERO DE
PUNTOS DE LA FFT
```

```
sli_N=icontrol(fig,'Style','slider','Position',[40          110          120
20],'SliderStep',[0.0333333333333333
0.1],'Min',128,'Max',2048,'Value',1024,...
```

```
'Callback','set(N_cur,"String",num2str(get(sli_N,"Val"))),N=get(sli_N,"Val");
ScriptespectroECG');
```

```
% Definir los controles de texto para los valores minimos
```

```
N_min=icontrol(fig,...
    'Style','text',...
    'Pos',[10 110 30 20],...
    'String',num2str(get(sli_N,'Min')));
```

```
% Definir los controles de texto para los valores maximos
```

```
N_max=icontrol(fig,...
    'Style','text',...
    'Pos',[160 110 30 20],...
    'String',num2str(get(sli_N,'Max')),...
    'Horiz','right');
```

```
% Definir las etiquetas de las barras
```

```
N_label=icontrol(fig,...
    'Style','text',...
    'Pos',[40 130 69 20],...
    'String','N-FFT');
```

```

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
N_cur=icontrol(fig,...
    'Style','text',...
    'Pos',[110 130 50 20],...
    'String',num2str(get(sli_N,'Value')));

% B. DEFINICION DE LA BARRA QUE CONTROLA EN NUMERO DE
MUESTRAS

sli_Ns=icontrol(fig,'Style','slider','Position',[40          210          120
20],'SliderStep',[0.001001001
0.1001001001],'Min',100,'Max',100000,'Value',2000,...

'CallBack','set(Ns_cur,"String",num2str(get(sli_Ns,"Val"))),Ns=get(sli_Ns,"V
al");ScriptespectroECG');

% Definir los controles de texto para los valores minimos
Ns_min=icontrol(fig,...
    'Style','text',...
    'Pos',[10 210 30 20],...
    'String',num2str(get(sli_Ns,'Min')));

% Definir los controles de texto para los valores maximos
Ns_max=icontrol(fig,...
    'Style','text',...
    'Pos',[160 210 35 20],...
    'String',num2str(get(sli_Ns,'Max')),...
    'Horiz','right');

% Definir las etiquetas de las barras
Ns_label=icontrol(fig,...

```

```

'Style', 'text',...
'Pos',[40 230 69 20],...
'String','MUESTRAS');

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
Ns_cur=icontrol(fig,...
'Style', 'text',...
'Pos',[110 230 50 20],...
'String',num2str(get(sli_Ns,'Value')));

% C. DEFINICION DE LA BARRA QUE CONTROLA TAMAÑO DE
VENTANA (WINDOW)

sli_Ws=icontrol(fig,'Style','slider','Position',[40          10          120
20],'SliderStep',[0.0025062656641604
0.25062656641604],'Min',128,'Max',51200,'Value',1024,...

'CallBack','set(Ws_cur,"String",num2str(get(sli_Ws,"Val"))),Ws=get(sli_Ws,'
'Val'));ScriptespectroECG');

% Definir los controles de texto para los valores minimos
Ws_min=icontrol(fig,...
'Style','text',...
'Pos',[10 10 30 20],...
'String',num2str(get(sli_Ws,'Min')));

% Definir los controles de texto para los valores maximos
Ws_max=icontrol(fig,...
'Style','text',...
'Pos',[160 10 35 20],...
'String',num2str(get(sli_Ws,'Max')),...
'Horiz','right');

```

```

% Definir las etiquetas de las barras
Ws_label=icontrol(fig,...
    'Style', 'text',...
    'Pos',[40 30 69 20],...
    'String','WINDOW');

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
Ws_cur=icontrol(fig,...
    'Style', 'text',...
    'Pos',[110 30 50 20],...
    'String',num2str(get(sli_Ns,'Value')));

% D. DEFINICION DE LA BARRA QUE CONTROLA DESPLAZAMIENTO
DE VENTANA DE DATOS

sli_Shift=icontrol(fig,'Style','slider','Position',[40          60          120
20],'SliderStep',[0.002 0.1],'Min',0,'Max',50000,'Value',0,...

'CallBack','set(Shift_cur,"String",num2str(get(sli_Shift,"Val"))),Shift=get(sli_
Shift,"Val");ScriptespectroECG');

% Definir los controles de texto para los valores minimos
Shift_min=icontrol(fig,...
    'Style','text',...
    'Pos',[10 60 30 20],...
    'String',num2str(get(sli_Shift,'Min')));

% Definir los controles de texto para los valores maximos
Shift_max=icontrol(fig,...
    'Style','text',...
    'Pos',[160 60 35 20],...
    'String',num2str(get(sli_Shift,'Max')),...
    'Horiz','right');

```

```

% Definir las etiquetas de las barras
Shift_label=icontrol(fig,...
    'Style', 'text',...
    'Pos',[40 80 69 20],...
    'String','SHIFT');

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
Shift_cur=icontrol(fig,...
    'Style', 'text',...
    'Pos',[110 80 50 20],...
    'String',num2str(get(sli_Shift,'Value')));

% E. DEFINICION DE LA BARRA QUE CONTROLA ZOOM A LA
VENTADA DATOS EN EL TIEMPO

sli_ZT=icontrol(fig,'Style','slider','Position',[40          160          120
20],'SliderStep',[0.001 0.1],'Min',0,'Max',100,'Value',100,...

'CallBack','set(ZT_cur,"String",num2str(get(sli_ZT,"Val"))),zoomt=get(sli_Z
T,"Val")/100;ScriptespectroECG');

% Definir los controles de texto para los valores minimos
ZT_min=icontrol(fig,...
    'Style','text',...
    'Pos',[10 160 30 20],...
    'String',num2str(get(sli_ZT,'Min')));

% Definir los controles de texto para los valores maximos
ZT_max=icontrol(fig,...
    'Style','text',...
    'Pos',[160 160 35 20],...

```

```

'String',num2str(get(sli_ZT,'Max'))),...
'Horiz','right');

% Definir las etiquetas de las barras
ZT_label=uicontrol(fig,...
'Style','text',...
'Pos',[40 180 69 20],...
'String','ZOOM');

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
ZT_cur=uicontrol(fig,...
'Style','text',...
'Pos',[110 180 50 20],...
'String',num2str(get(sli_ZT,'Value')));

% II.- CONTROLES DE PANTALLA DE ANALIZADOR DE ESPECTROS
%
% A.- DEFINICION DE LA BARRA QUE CONTROLA LA ZOOM DE
ESCALA DE FRECUENCIA

sli_a2=uicontrol(fig,'Style','slider','Position',[250          10          120
20],'SliderStep',[0.001 0.1],'Min',0,'Max',100,'Value',100,...

'CallBack','set(a2_cur,"String",num2str(get(sli_a2,"Val"))),zoomf=get(sli_a2
,"Val")/100;ScriptespectroECG');

% Definir los controles de texto para los valores minimos
a2_min=uicontrol(fig,...
'Style','text',...
'Pos',[220 10 30 20],...
'String',num2str(get(sli_a2,'Min')));

```

```

% Definir los controles de texto para los valores maximos
a2_max=icontrol(fig,...
    'Style','text',...
    'Pos',[370 10 30 20],...
    'String',num2str(get(sli_a2,'Max')),...
    'Horiz','right');

% Definir las etiquetas de las barras
a2_label=icontrol(fig,...
    'Style', 'text',...
    'Pos',[250 30 69 20],...
    'String','ZOOM FREC');

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
a2_cur=icontrol(fig,...
    'Style', 'text',...
    'Pos',[320 30 50 20],...
    'String',num2str(get(sli_a2,'Value')));

% B.- DEFINICION DE LA BARRA QUE CONTROLA EL UMBRAL DE
% AMPLITUD XF, ZOOM ESCALA XF

sli_XF=icontrol(fig,'Style','slider','Position',[250          110          120
20],'SliderStep',[0.001 0.1],'Min',0,'Max',100000,'Value',10000,...

'CallBack','set(XF_cur,"String",num2str(get(sli_XF,"Val"))),umbral_XF=get(
sli_XF,"Val");ScriptespectroECG);

% Definir los controles de texto para los valores minimos
XF_min=icontrol(fig,...
    'Style','text',...
    'Pos',[220 110 30 20],...

```

```

'String',num2str(get(sli_XF,'Min')));

% Definir los controles de texto para los valores maximos
XF_max=icontrol(fig,...
    'Style','text',...
    'Pos',[370 110 35 20],...
    'String',num2str(get(sli_XF,'Max')),...
    'Horiz','right');

% Definir las etiquetas de las barras
XF_label=icontrol(fig,...
    'Style', 'text',...
    'Pos',[250 130 69 20],...
    'String','ZOOM MAGX');

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
XF_cur=icontrol(fig,...
    'Style', 'text',...
    'Pos',[320 130 50 20],...
    'String',num2str(get(sli_XF,'Value')));

% III.- CONTROLES DE PANTALLA DE ESPECTROGRAMA

%A.-  DEFINICION DE LA BARRA QUE CONTROLA ZOOM DE
FRECUENCIA

sli_a3=icontrol(fig,'Style','slider','Position',[460          10          120
20],'SliderStep',[0.001 0.1],'Min',0,'Max',200,'Value',100,...

'CallBack','set(a3_cur,"String",num2str(get(sli_a3,"Val"))),zoomFre=get(sli_
a3,"Val")/100;ScriptespectroECG');

% Definir los controles de texto para los valores minimos

```

```

a3_min=icontrol(fig,...
    'Style','text',...
    'Pos',[430 10 30 20],...
    'String',num2str(get(sli_a3,'Min')));

% Definir los controles de texto para los valores maximos
a3_max=icontrol(fig,...
    'Style','text',...
    'Pos',[580 10 30 20],...
    'String',num2str(get(sli_a3,'Max')),...
    'Horiz','right');

% Definir las etiquetas de las barras
a3_label=icontrol(fig,...
    'Style', 'text',...
    'Pos',[460 30 69 20],...
    'String','ZOOM FREC');

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
a3_cur=icontrol(fig,...
    'Style', 'text',...
    'Pos',[530 30 50 20],...
    'String',num2str(get(sli_a3,'Value')));

%B.- DEFINICION DE LA BARRA QUE CONTROLA ZOOM DE ESCALA
TIEMPO

sli ST=icontrol(fig,'Style','slider','Position',[460          110          120
20],'SliderStep',[0.001 0.1],'Min',0,'Max',100,'Value',100,...

'CallBack','set(ST_cur,"String",num2str(get(sli_ST,"Val"))),zoomST=get(sli_
ST,"Val")/100;ScriptespectroECG');

```

```

% Definir los controles de texto para los valores minimos
ST_min=icontrol(fig,...
    'Style','text',...
    'Pos',[430 110 30 20],...
    'String',num2str(get(sli_ST,'Min')));

% Definir los controles de texto para los valores maximos
ST_max=icontrol(fig,...
    'Style','text',...
    'Pos',[580 110 30 20],...
    'String',num2str(get(sli_ST,'Max')),...
    'Horiz','right');

% Definir las etiquetas de las barras
ST_label=icontrol(fig,...
    'Style', 'text',...
    'Pos',[470 130 69 20],...
    'String','ZOOM TIEMP');

% Definir los controles de texto para los valores actuales
% Los valores son inicializados aqui y son modificados por los callbacks
% de las barras cuando el usuario cambia sus valores
ST_cur=icontrol(fig,...
    'Style', 'text',...
    'Pos',[530 130 50 20],...
    'String',num2str(get(sli_ST,'Value')));

%C.-  DEFINICION DE LA BARRA QUE CONTROLA ZOOM DE
AMPLITUD, MATRIZ B

sli_B=icontrol(fig,'Style','slider','Position',[460          210          120
20],'SliderStep',[0.001 0.1],'Min',0,'Max',100000,'Value',40000,...

'CallBack','set(B_cur,"String",num2str(get(sli_B,"Val"))),umbral_B=get(sli_B
,"Val");ScriptespectroECG');

```

```
% Definir los controles de texto para los valores minimos
```

```
B_min=icontrol(fig,...  
    'Style','text',...  
    'Pos',[430 210 30 20],...  
    'String',num2str(get(sli_B,'Min')));
```

```
% Definir los controles de texto para los valores maximos
```

```
B_max=icontrol(fig,...  
    'Style','text',...  
    'Pos',[580 210 35 20],...  
    'String',num2str(get(sli_B,'Max')),...  
    'Horiz','right');
```

```
% Definir las etiquetas de las barras
```

```
B_label=icontrol(fig,...  
    'Style', 'text',...  
    'Pos',[470 230 69 20],...  
    'String','ZOOM MAGB');
```

```
% Definir los controles de texto para los valores actuales
```

```
% Los valores son inicializados aqui y son modificados por los callbacks
```

```
% de las barras cuando el usuario cambia sus valores
```

```
B_cur=icontrol(fig,...  
    'Style', 'text',...  
    'Pos',[530 230 50 20],...  
    'String',num2str(get(sli_B,'Value')));
```

C.2.2. Módulo Espectrógrafo del ECG-EEG-SISM.-

Script:

```

%*****
%***** ANALIZADOR DE ESPECTROS DIGITAL *****
%*****
% SCRIPT PARA IMPLEMENTAR EL ANALIZADOR DE ESPECTROS
DIGITAL
% BASADO EN LA APLICACION DE LA FFT
%*****
% MODULO DE CONTROL DE INTERFAZ GRAFICA USUARIO
% AUTOR: OSCAR TORRES MEJIA
% FECHA: FEBRERO 2005
% FILE : ScriptespectroECG.m
% V CURSO DE TITULACION FIEE
%*****
%
%
% MODULO DE ADQUISICION DE DATOS. Lee datos almacenado en
archivos segun
% opcion elegida
switch opcion
case 1
    ft_dir7_label=uicontrol(fig,...
        'Style', 'text',...
        'BackgroundColor','White',...
        'FontSize', 16,...
        'FontWeight','bold',...
        'HorizontalAlignment','center',...
        'Pos',[10 350 610 50],...
        'String','ANALISIS SEÑAL ECG');

    Fs=100;
    idf=fopen('ECGPacO3full.txt');
    ferror(idf);
    [Si,COUNT]=fscanf(idf,'%5g',Ns);
    st=fclose(idf);
case 2
    ft_dir7_label=uicontrol(fig,...
        'Style', 'text',...
        'BackgroundColor','White',...
        'FontSize', 16,...
        'FontWeight','bold',...
        'HorizontalAlignment','center',...
        'Pos',[10 350 610 50],...
        'String','ANALISIS SEÑAL EEG');

    Fs=100;

```

```

%     idf=fopen('EEGPacO5full1ch.txt');
idf=fopen('EEGPacO3full2ch.txt');
ferror(idf);
[Si,COUNT]=fscanf(idf,'%f',Ns);
st=fclose(idf);
case 3
ft_dir7_label=icontrol(fig,...
'Style','text',...
'BackgroundColor','White',...
'FontSize',16,...
'FontWeight','bold',...
'HorizontalAlignment','center',...
'Pos',[10 350 610 50],...
'String','ANALISIS SEÑAL SISMICA');

Fs=200;
%idf=fopen('tac2-0405301144fEW.txt');
idf=fopen('moq1-0106231533f-EW.txt');
%idf=fopen('csm-0407290500fa1.txt');
ferror(idf);
[Si,COUNT]=fscanf(idf,'%f',Ns);
st=fclose(idf);
case 4
ft_dir7_label=icontrol(fig,...
'Style','text',...
'BackgroundColor','White',...
'FontSize',16,...
'FontWeight','bold',...
'HorizontalAlignment','center',...
'Pos',[10 350 610 50],...
'String','ANALISIS SEÑAL PATRON');

% Senales senoidales patron
Fs=200;
tt=0:(1/Fs):((Ns-1)/Fs);
f=1;
Si=cos(2*pi*tt*f)+cos(2*pi*tt^2*f)+cos(2*pi*tt^4*f);
otherwise
ft_dir7_label=icontrol(fig,...
'Style','text',...
'BackgroundColor','Red',...
'FontSize',16,...
'FontWeight','bold',...
'HorizontalAlignment','center',...
'Pos',[10 350 610 50],...
'String','ERROR:SELECCIONAR OPCION');
end;

Ts=1/Fs ; % periodo de señal de muestreo, o periodo entre muestreos

% Comprobacion de Valores de ventana y cantidad de muestras

```

```

if (Ws+Shift)<=Ns
    if Ws>=N
        ft_dir6_label=uicontrol(fig,...
            'Style', 'text',...
            'BackgroundColor','Green',...
            'FontSize',12,...
            'FontWeight','bold',...
            'HorizontalAlignment','center',...
            'Pos',[10 300 610 50],...
            'String','PARAMETROS CORRECTOS');
    else
        ft_dir6_label=uicontrol(fig,...
            'Style', 'text',...
            'BackgroundColor','Red',...
            'FontSize',12,...
            'FontWeight','bold',...
            'HorizontalAlignment','center',...
            'Pos',[10 300 610 50],...
            'String','ERROR: VENTANA DEBE SER MAYOR');
    end;
else
    ft_dir6_label=uicontrol(fig,...
        'Style', 'text',...
        'BackgroundColor','Red',...
        'FontSize',12,...
        'FontWeight','bold',...
        'HorizontalAlignment','center',...
        'Pos',[10 300 610 50],...
        'String','ERROR: DESBORDE DE VENTANA');
end;

% Umbral Xf, para zoom de pantalla de analizador de espectros

%Limpia la matriz auxiliar Magx1
clear S;

shift=round(Shift);

% Modulo que Desplaza senal Shift posiciones en la cadena de datos
% muestreados

for ij=1:1:Ws
    S(ij)=Si(ij+Shift);
end;

% Definicion de rango de tiempo para la señal a analizar
t=0:Ts:((Ws-1)/Fs);
time=(Ws-1);
subplot(2,2,1),
plot(t(1:round(time*zoomt)),S(1:round(time*zoomt))),title('Señal'),
xlabel('kT,seg');
ylabel('s(kT)');
grid;

```

```

% *****
% Redondeo de parametros seleccionados *
% *****

N=round(N);
Ns=round(Ns); %Definicion del numero de ondas de la mas baja frecuencia
Ws=round(Ws);

k=0:1:Ws-1;

% *****
% Usando la FTT simple *
% *****

F=fft(S);
Magx=abs(F);
hertz=k*(1/(Ws*Ts));

% Umbral Xf, para zoom de pantalla de analizador de espectros

%Limpia la matriz auxiliar Magx1
clear Magx1;

% Modulo que filtra
for i=1:1:Ws
    if Magx(i)<=umbral_XF
        Magx1(i)=Magx(i);
    else
        Magx1(i)=umbral_XF;
    end;
end;

%subplot(2,2,2),stem(hertz(1:round(Ns*res)),Magx(1:
round(Ns*res))),title('Magnitud de Xf'),
%xlabel('Hz'),
%grid;
subplot(2,2,2),plot(hertz(1:round((Ws/2)*zoomf)),Magx1(1:
round((Ws/2)*zoomf))),title('Magnitud de Xf'),
xlabel('Hz'),
grid;

% *****
% ***** ESPECTROGRAMA *****
% *****

ww=hamming(N);
subplot(2,2,4),
[B, Fre, Tiem]=specgram(S,N,Fs,ww);

%Se define control de umbral, para eliminar las señales erroneas en frec=0
%a traves de un valor umbral

%Limpia la matriz auxiliar B1
clear B1;

```

```
% Se evalua cada valor de la matriz B, truncandolo a Limite_B, para mejorar
la
% visualizacion de las senales de menor amplitud o potencia, y asi el
% efecto de control de zoom
for ii=1:1:length(Tiem)
    for jj=1:1:length(Fre)
        if abs(B(jj,ii))<=umbral_B
            B1(jj,ii)=B(jj,ii);
        else
            B1(jj,ii)=umbral_B;
        end;
    end,
end;

imagesc(Tiem,Fre,abs(B1)), axis xy;
axis ([0 max(t*zoomST) 0 max(Fre*zoomFre)]), colormap(jet);
colorbar('vert');
xlabel('Tiempo(Seg)'),ylabel('Frecuencia(Hz)');
abcd = sprintf('ESPECTROGRAMA');
set(get(gca,'title'),'String',abcd);
zoom on;
grid;
```

ANEXO D

MANUAL DE USO DEL INSTRUMENTO: ANALIZADOR DE ESPECTROS DIGITAL FFT

Pantalla de Visualización General.-

Es un módulo que constituye la interfaz gráfica con el usuario, la misma que ocupa todo el espacio de la pantalla de la PC y en el que se visualizan:

1. Barra de control General.-

Barra con iconos ubicada en la línea superior izquierda de la pantalla y que contiene los comandos típicos del windows, en la que se ha habilitado adicionalmente la función menú específica denominada 'Opcion'.

Descripción de menú de 'Opcion'.- Dentro de las opciones de menú activas en el instrumento, se ha creado la opción de menú 'Opcion', y dentro de él, las sub-opciones ECG, EEG, SISIM, y PATRON, con las cuales se elige que tipo de señal que se ha de analizar en particular: ECG, para el electrocardiograma, EEG, para el electroencefalograma, SISIM, para el análisis de señales sísmicas y PATRON, que permite generar frecuencias patrón y poder comprobar el correcto funcionamiento del instrumento.

Advertencia: Para poder operar el instrumento, es forzoso la elección de la opción deseada, de otra forma saldrá error en la ejecución del programa.

2. El Tablero de Control del instrumento.-

Sub-panel, contenido dentro del panel mayor, y contiene a su vez:

- Cuadro con el **Nombre** del Instrumento: “**ANALIZADOR DE ESPECTROS**”.
- Cuadro con el nombre del **Tipo de Señal** que se está analizando, la cual fue elegida en la barra de menú.
- **Cuadro de mensajes.-** Que indican dos cosas: (1) “PARAMETROS CORRECTOS”, cuando el aplicado ha verificado que las condiciones del tamaño de muestra, el tamaño de la ventana y el N-FFT son los adecuados. (2) “ERROR: VENTANA DEBE SER MAYOR”, que se presenta cuando el tamaño de la ventana de muestras es menor que la exigida por el N-FFT para una correcta evaluación.
- **Sub-tablero de control de Parámetros Generales.-** El que tiene como finalidad el calibrar los valores de parámetros generales del instrumento, entre estos podemos indicar los siguientes:
 - **El Número de Elementos de Muestra.-** Los valores para éste parámetro son definidos por una barra de control deslizante, y determina el número de muestras que se extraerán del archivo base que contiene los datos que podrían ser analizados. Los valores mínimos son ‘0’ y el máximo ‘100,000’. Cuando se indica esto, es porque no todos los datos definidos por este parámetro serán analizados, sino únicamente los que sean determinados por la ventana de muestras.
 - **Zoom de la Señal en función del tiempo.-** Control deslizante que permite hacer un acercamiento a la señal en función del tiempo. El valor mínimo es ‘0’ y el máximo es ‘100’, representando el 100% de la imagen de la señal en función del tiempo. Este control facilita hacer un acercamiento de la señal, para poder apreciarla mejor para valores mínimos en la escala del tiempo.
 - **N-FFT.-** Número de puntos de evaluación de la Transformada Rápida de Fourier, cuyo valor es determinado por una barra de control deslizable. El valor

mínimo se ha establecido en 128 y el máximo en 2048, y los pasos entre cada desplazamiento se ha fijado en pasos de 64.

- **Desplazamiento de la ventana o Shift.-** La función de esta barra deslizante es desplazar la ventana de muestras en un valor fijado por este control, lo cual permite explorar con la ventana, datos que están contenidos en el registro mayor de muestras a analizar, y escoger dentro de ella la sección particular a analizar. Los valores mínimo y máximo son '0' hasta '50,000'

- **El tamaño de la ventana de muestras.-** La función de esta barra de control deslizante es determinar la ventana de análisis de muestras. Esto es, fijar un rango de datos del registro de la muestra total, y que se tiene particular interés en analizar. Este control está asociado al control de shift o desplazamiento de ventana. Los valores definidos mínimo '128' hasta un máximo de '100,000'.

- **Sub-tablero Respuesta en Frecuencia.-** El que tiene como finalidad el calibrar elementos de visualización del analizador de espectros o espectrógrafo, entre estos podemos indicar los siguientes:
 - **ZOOM MAGX.-** Los valores para éste parámetro son definidos por una barra de control deslizante, y determina el valor máximo en magnitud de la señal X_f , resultante de aplicar la FFT. La función de este control es deslizar un valor umbral con el que se permita visualizar señales que estén debajo de ese valor umbral. Entre las aplicaciones de este control está el caso de una señal con una gran componente de frecuencia cero, impedirá que las componentes de otras frecuencias se visualicen reducidas, pero a través del desplazamiento de este control, será posible ir seleccionando las amplitudes del espectro en frecuencia de menos potencias que requieran ser analizadas. Sus valores están definidos desde '0' hasta '100,000'

 - **ZOOM FREC.-** Este control deslizante permite aplicar un efecto de acercamiento en función de la frecuencia del gráfico resultante del análisis de frecuencia FFT. Sus valores están definidos entre '0%' al '100%'.

- **Sub-tablero Espectrograma.-** El que tiene como finalidad el calibrar elementos de visualización del espectrograma, entre estos podemos indicar los siguientes:

- **ZOOM MAG.-** Los valores para éste parámetro son definidos por una barra de control deslizante, y determina el valor máximo en magnitud de la señal B, resultante de aplicar la 'espec'. La función de este control es deslizar un valor umbral con el que se permita visualizar señales que tengan un nivel de potencia que estén debajo de ese valor umbral. Entre las aplicaciones de este control está el caso de una señal con una gran componente de frecuencia cero, impedirá que las componentes de otras frecuencias se visualicen reducidas, pero a través del desplazamiento de este control, será posible ir seleccionando las amplitudes del espectro en frecuencia de menos potencias que requieran ser analizadas. Sus valores están definidos desde '0' hasta '100,000'
- **ZOOM TIEM.-** Este control deslizante permite aplicar un efecto de acercamiento en función del tiempo en el espectrograma. Permite ver con más claridad la respuesta en el tiempo de la señal, que para el valor normal (100%) no sería factible visualizar adecuadamente. Sus valores están definidos entre '0%' al '100%'.
- **ZOOM FREC.-** Este control deslizante permite aplicar un efecto de acercamiento en función de la frecuencia del gráfico resultante del espectrograma. Permite ver con más claridad la respuesta en frecuencia de la señal, que para el valor normal (100%), no sería factible visualizar adecuadamente. Sus valores están definidos entre '0%' al '100%'.

3. El cuadro del diagrama de la Señal en función del tiempo.-

Cuadro ubicado en la parte superior izquierda del panel principal, y permite la visualización del diagrama de amplitud de la señal analizada en función del tiempo. Sobre ésta influyen los controles de parámetros generales, excepto el N-FTT, dado que expresa la respuesta en dominio del tiempo de la señal analizada.

4. El espectrógrafo digital.-

Cuadro que se ubica en la parte superior derecha del panel general y que visualiza el resultado del análisis FFT de la señal, en la se muestran las amplitudes

Xf (el nivel de potencia de la señal) en función de la frecuencia. Es decir, este diagrama muestra la respuesta en frecuencia de la señal, y para esto influyen todos los controles del panel, y,

5. El espectrograma.-

Cuadro que se ubica en la parte inferior derecha del panel general y que visualiza el análisis de la señal en la frecuencia y el comportamiento de la frecuencia en relación al tiempo, simultáneamente. Es decir, se tiene dentro de ciertos límites, la diagramación en ambos dominios, en el tiempo y en la frecuencia de la señal analizada.

BIBLIOGRAFÍA

1. Alan V. Oppenheim & Alan S. Willsky, "Señales y Sistemas", Prentice Hall, 1994.
2. John G. Proakis & Dimitris G. Manolakis, "Tratamiento digital de Señales", Prentice Hall, 1998.
3. Alan V. Oppenheim & Ronald W. Schafer, "Tratamiento de Señales en Tiempo Discreto", Prentice Hall, 2000.
4. Emmanuel C. Ifeachor & Barrie W. Jervis, "Digital Signal Processing – A Practical Approach", Addison-Wesley, 1993.
5. Theory of the DFT, <http://www.spd.eee.strath.ac.uk/~interact/fourier/dft.html>, extraída el 19 de Diciembre del 2004.
6. Theory of the DFT, http://en.wikipedia.org/wiki/Discrete_Fourier_Transform, extraída el 12 de Diciembre del 2004.
7. Theory of the FFT, <http://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/transform/fft.html>, extraída el 13 de Noviembre del 2004.
8. Theory of the FFT, <http://astron.berkeley.edu/~ngst/fft/fft.html>, extraída el 20 de Noviembre del 2004.
9. Standard 12 Lead ECG, http://medstat.med.utah.edu/kw/ecg/ecg_outline/, extraída el 12 de Diciembre del 2004.

10. Triángulo de Eindhoven en ECG, http://medstat.med.utah.edu/kw/ecg/ecg_outline/, extraída el 12 de Diciembre de 2004.
11. Ondas EEG, http://www.brown.edu/Departments/Clinical_Neurosciences/louis/eegcrs.html, extraída el 12 de Diciembre del 2004.
12. Registros sísmicos Cismid, http://www.cismid-uni-org/p_acelerograf/registros2.htm, extraído el 18 de Diciembre del 2004.