UNIVERSIDAD LA SALLE

ESCUELA DE INGENIERIA

INCORPORADA A LA U.N.A.M.

DESARROLLO, COMPARACION Y USO DE LAS TRANSFORMADAS: FOURIER, HARTLEY, WALSH, HAAR Y EL METODO DE MAXIMA ENTROPIA PARA EL ANALISIS DE ESPECTRO

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A

EDUARDO GOMEZ RAMIREZ

ASESOR DE TESIS: ING. GUILLERMO ARANDA PEREZ

MEXICO, D. F. 1993

DESARROLLO, COMPARACION Y USO DE LAS TRANSFORMADAS: FOURIER, HARTLEY, WALSH, HAAR Y EL METODO DE MAXIMA ENTROPIA PARA EL ANALISIS DE ESPECTRO

CONTENIDO

DESARROLLO, COMPARACION Y USO DE LAS TRANSFORMADAS: FOURIER, HARTLEY , WALSH, HAAR Y EL METODO DE MAXIMA ENTROPIA PARA EL ANALISIS DE ESPECTRO

INTRODUCCION.

" I want to know how God created this world . I am not interested in this or that phenomenon, in the **spectrum** of this or that element.
I want to know His thoughts; the rest are details "

ALBERT EINSTEIN.

Uno de los primeros trabajos en el área de Análisis de Espectro son los desarrollados por el barón Jean-Baptiste Joseph Fourier[1] (1768-1830) para la descripción de distribución de temperaturas. Fourier tenía una pasión por el calor tal, que se decía que mantenía su casa de Grenoble en tal temperatura que resultaba incómoda para sus visitas. Quizá una de las causas fue el atractivo de climas cálidos lo que, en 1798, indujo a Fourier a unirse a la comitiva de 165 sabios que acompañaron la expedición de Napoleón a Egipto.

Mientras Napoleón combatía en Palestina, y expulsaba de Egipto a los turcos y perseguía a Murad Bey, jefe de los mamelucos, los científicos franceses emprendieron ambiciosos estudios de Geografía, Arqueología, Medicina, Agricultura e Historia Natural. Fourier fue nombrado secretario del organismo científico conocido por Instituto de Egipto y por su gran capacidad en tareas administrativas, le fueron encomendadas varias misiones diplomáticas, lo cual no le impidió llevar a cabo una exhaustiva

[1] Antes de Fourier, ya se tenía idea de la "descomposición" de funciones: Claudio Ptolomeo (S.II) y Euler (1707-1783) (Bracewell,1986)

investigación de las antigüedades egipcias y el desarrollo de una teoría sobre las raíces de ecuaciones algebraicas.

Poco antes de que los franceses fueran arrojados de Egipto, en 1801, Fourier y sus colegas se embarcan para volver a Francia. Pero el comandante de la Flota Británica, Almirante Sir Sidney Smith no tardó en apoderarse del navío y de su cargamento de documentación y reliquias egipcias. Todos los científicos desembarcaron en Alejandría, y tiempo después se devolvió el material confiscado a Francia con excepción de la piedra Rosetta (la clave de los jeroglífico egipcios) que se conserva todavía hoy en el Museo Británico, como monumento a la derrota a militar de Napoleón y su contribución a la Egiptología.

Al regresar a Francia, Fourier se dedicó a cuestiones matemáticas, en su puesto de profesor de análisis de la Escuela Politécnica, aunque en el año de 1802 vuelve al servicio de Napoleón. Fourier se convierte en prefecto del departamento de Isére. Durante ese tiempo, deduce una ecuación que describía la conducción del calor en los cuerpos sólidos. Y en el año de 1807, desarrolla un método para resolver tal ecuación: **LA TRANSFORMADA DE FOURIER.**

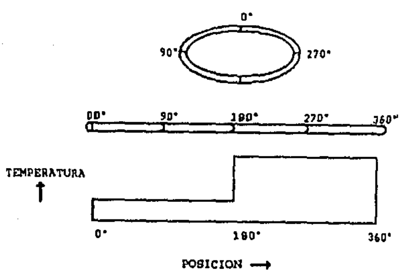


Figura 1 Distribución de calor en un anillo

El utiliza esta técnica para explicar muchos ejemplos de conducción del calor. Por ejemplo el flujo de calor en torno al anillo de un ancla (un anillo de hierro que sujeta el ancla de un barco a su cadena) que se introduce a la mitad en el fuego. Cuando parte de la circunferencia del anillo se pone al rojo vivo, éste se retira y antes de que pierda el calor obtenido se entierra en arena refractaria fina y se mide la temperatura en torno a la curva exterior. (Bracewell,1989) (fig. 1).

La distribución de temperatura al principio es irregular, esto es, parte del anillo se encuentra uniformemente frío y parte uniformemente caliente, en la zona media, la temperatura cambia bruscamente. Sin embargo, debido a la transmisión de calor desde la parte caliente hacia la parte fría, la distribución de temperatura comienza a suavizarse. La distribución de temperaturas en torno al anillo alcanza pronto una forma senoidal; esto es, al representar gráficamente la temperatura, se puede observar una onda que sube y baja suavemente, de forma exactamente igual que el de las funciones seno y coseno. Esta senoide se va aplanando gradualmente, hasta que todo el anillo alcanza una temperatura constante.

Fourier propuso que la distribución irregular inicial podía descomponerse en multitud de senoidales simples, donde cada una tenía una amplitud correspondiente a su propia temperatura máxima y fase, es decir, su posición relativa dentro del anillo.

Además, cada componente senoidal variaba un número entero de veces de un máximo a un mínimo e inversamente en cada vuelta completa en torno al anillo. La variación que poseía un solo ciclo se le llamó *armónico fundamental*, mientras que las variaciones con dos, tres o más ciclos por giro en torno al anillo son el *segundo armónico, tercer armónico*, etc..... La función matemática que describe la temperatura máxima y la posición (fase) de cada uno de los armónicos es la Transformada de Fourier de la distribución de temperaturas. Fourier había cambiado una distribución única, cuya descripción matemática era difícil, por una serie más "manejable" de funciones trigonométricas periódicas que, al sumarse reconstruyen la distribución original.

Al aplicar el análisis anterior a la conducción del calor en torno al anillo, razonó que, cuanto mayor sea el número de períodos de una componente senoidal, con mayor rapidez se atenuará tal componente. Esto puede ser explicado si se observa que la temperatura del segundo armónico varía dos veces de caliente a fría al ir recorriendo la circunferencia del anillo, mientras que la del armónico fundamental lo hace solamente una vez; de esta manera, la distancia que debe viajar el calor desde la cresta térmica hasta el valle frío es, para el segundo armónico, la mitad de la que corresponde a la fundamental. Además el gradiente de temperatura es, en el segundo armónico, el doble de abrupto que en la variación fundamental. Dado que un flujo calorífico doble ocupa la mitad de la distancia, el segundo armónico se extinguirá cuatro veces antes que el fundamental.

El análisis de Fourier ponía en entredicho ciertas concepciones matemáticas que sus contemporáneos tomaban como dadas. A principios del siglo XIX, a muchos de los más distinguidos matemáticos parisienses (Lagrange, Laplace, Legendre, Biot, Poisson) les resultaba difícil aceptar la tesis de Fourier, que sostenía que la distribución de temperaturas podía descomponerse en una sencilla suma aritmética, compuesta por una variación fundamental más sus armónicas de frecuencias más elevadas. También Leonhard Euler halló incorrectas las ideas de Fourier, a pesar de que él mismo propuso que ciertas funciones eran representables mediante sumas de funciones senoidales. y se opuso a esta idea cuando Fourier presentó su tesis en la Academia Francesa de Ciencias.

La gran desconfianza con que los colegas de Fourier consideraban su trabajo, causó que su publicación se retrasara hasta 1815. De hecho, el trabajo no quedó plenamente descrito hasta la publicación, en 1822, de su libro "Théorie analytique de la chaleur" (Teoría analítica del calor).

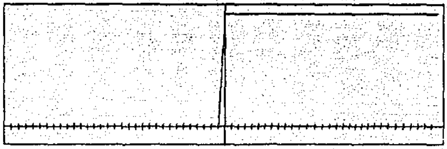


Fig. 2 Función Escalón

Las objeciones al método de Fourier se centraban en su proposición de que una función en apariencia discontinua pudiera representarse mediante una suma de funciones senoidales continuas, por ejemplo como la función de Heaviside (función escalón,fig. 2).

Los matemáticos de ésta época, jamás habían visto que una función discontinua estuviera descrita por una combinación de funciones lineales,cuadráticas, exponenciales o senoidales, y la idea de que la suma de un número infinito de senoidales podía ser convergente, y representar con exactitud funciones que tuvieran saltos bruscos, les parecía absurda.

A pesar de la polémica generada, algunos investigadores como la matemática Sophie Germain y el Ingeniero Claude Navier, comenzaron a generalizar el trabajo de Fourier en otras áreas distintas a la del análisis del calor, y así poco a poco la cantidad de áreas en las cuales la Transformada de Fourier se iba aplicando, aumentaba paralelamente de la confirmación teórica de la convergencia para funciones discontinuas.

A partir de ese momento, las áreas y aplicaciones en las cuales La Transformada de Fourier era útil, iba aumentando, al mismo tiempo que la cantidad de datos y operaciones que se requerían para su obtención. Debido a esto, hubo un gran desarrollo en nuevos algoritmos que hicieran este análisis. Uno de ellos fue elaborado en 1965, por James W. Cooley y John W. Tukey. El trabajo de ambos dió lugar a un programa conocido por **Transformada Rápida de Fourier** (Fast Fourier Transform, FFT). Este algoritmo realiza la transformación en menor tiempo que la Transformada

Discreta de Fourier, por medio de una factorización de las operaciones que se repiten dentro del cálculo, reduciendo el número de multiplicaciones necesarias.

Hoy en día, existen diferentes algoritmos de la Transformada Rápida (Bones, 1987) como: "decimation in time", "decimation in frequency", Sande-Tukey, Cooley-Tukey que están implementados como funciones, subrutinas o instrucciones así como en procesadores de alta velocidad (Sophie, 1991), y se puede obtener información y bibliografía de las "1001 formas" de la Transformada de Fourier.

Pero este no es el único algoritmo utilizado (Raggi,1992) y en muchos casos la Transformada de Fourier no es la más óptima en cuanto análisis de señales se refiere. Existen otros algoritmos con un gran desarrollo teórico que empieza desde los trabajos de Heissenberg, Schrödinger, Von Neumann, Wiener, etc.. (Robinson, 1982) que son poco utilizados y en muchos casos se desconoce su existencia.

A pesar de que las referencias de varios artículos en México , es casi imposible conseguirlas, existe una gran cantidad de artículos que hacen referencia o que utilizan alguna Transformada diferente a la de Fourier[2], sumado a ésto, están los problemas de que en los artículos consultados o en algunos casos, libros, las fórmulas, programas o figuras están mal editados o "falta información" para su implementación (como sucedió en la mayor parte de los artículos principales de esta tesis). Otro problema que se encontró, fue el que existen pocos artículos que hagan una comparación de diferentes transformadas para tener una guía (que no sea "la experiencia") de cual es la transformada que se debe aplicar en el análisis de espectro.

En este trabajo se presenta una recopilación y planteamiento teórico para las transformadas: Fourier (FFT), Hartley (FHT), Walsh (FWT), Haar (FHAT), y Máxima entropía, comparación en cuanto a propiedades, desarrollo computacional (programas) , y un análisis de que

[2] Ver referencias

ventajas o desventajas tienen una sobre otra, dependiendo de la naturaleza de las señales a analizar.

La tesis está estructurada de la siguiente forma:

-En el capítulo I se hace una breve descripción sobre los fundamentos de las series ortogonales como base matemática de las transformaciones.

-En los capítulos II, III, IV, V y VI se desarrolla la teoría, propiedades, programa en computadora y algunas relaciones para las Transformadas de Fourier, Hartley, Walsh, Haar y el Método de Máxima Entropía.

- En el capítulo VII, se utiliza la teoría desarrollada anteriormente para comparar las transformadas y se presentan los espectros para algunas funciones.

CAPITULO I.- FUNDAMENTOS DE SERIES ORTOGONALES.

Una de las ideas más importantes en el análisis de espectro es la de representar una función f(t) como la suma de funciones más simples que faciliten el manejo o tratamiento de las señales; estas funciones forman lo que se llama una "**serie ortogonal**". Esto puede ser representado como (Beauchamp,1975):

$$f(t) = \int_0^T C_n * S_n(t) \qquad \text{Ec. 1.1}$$

donde: f(t) es una función del tiempo.
C_n es una constante que indica la magnitud de cada término de la serie.
$S_n(t)$ es una serie ortogonal.

La serie $S_n(t)$ (n = 0,1,2,3.....) se dice que es ortogonal con peso k sobre el intervalo $0 \geq t \geq T$ si (Zill, 1988):

$$\int k * S_n(t) * S_m(t) = \begin{cases} k & \text{si} \quad m = n \\ 0 & \text{si} \quad m \neq n \end{cases} \qquad \text{Ec. 1.2}$$

donde: n y m son valores enteros.
k es una constante independiente de n y m.

Los valores C_n deben ser tales que se minimice el error cuadrático medio E. C. M. ("Mean square error ", M.S.E.), de esta manera:

$$\text{M.S.E.} = \int_0^T \left[f(t) - \sum_{n=0}^{N-1} C_n * S_n(t) \right]^2 dt \qquad \text{Ec. 1.3}$$

y esto es posible haciendo C_n:

$$C_n = \frac{1}{T}\int_0^T f(t) * S_n(t)\, dt \qquad \text{Ec. 1.4}$$

El E.C.M. debe tender a cero cuando N tiende a infinito. Esto es posible cuando se utilizan series completas de funciones ortogonales, como lo son: seno, coseno, funciones de Haar, Walsh, etc.. Esto se expresa como:

$$\lim_{N \to \infty} \int_0^T \left[f(t) - \sum_{n=0}^{N-1} C_n * S_n(t) \right] dt = 0 \qquad \text{Ec. 1.5}$$

Cuando se utilizan series incompletas con funciones ortogonales, el error (M.S.E.) no converge y por lo tanto, no se puede representar cualquier tipo de función (aunque puede tener otras propiedades de igual importancia). Un ejemplo de este tipo de series incompletas son las Funciones de Rademacher.[1]

Como conclusión: "*Una función f(t) se puede representar como una serie de tiempo ortogonal (completa o incompleta), donde los coeficientes de la serie son obtenidos de la Ec. 1.4. De esta manera la función f(t) puede ser representada como un conjunto de coeficientes o números espectrales, que dependen únicamente del tipo de serie que se esté utilizando (Fourier, Hartley, Walsh, Haar, etc), y que se pueden graficar magnitud Cn vs. n, donde n es el enésimo término de la serie. A esta gráfica se le va a denominar* ***"espectro de f(t)"***.

[1] Ver apéndice 1.

CAPITULO II.- TRANSFORMADA DE FOURIER.

2.1 DEFINICIÓN

Para ejemplificar el planteamiento de la Ec. 1.2 considérense dos funciones tales que:

$$S_n(t) = \sqrt{2}\ \cos(2\pi nt)$$
$$S_m(t) = \sqrt{2}\ \cos(2\pi mt)$$

Sustituyendo en la Ec. 1.2 se tiene que:

$$= \int 2\cos(2\pi nt)\ \cos(2\pi mt)\,dt$$

$$= \int \cos[(m+n)2\pi t] + \cos[(m-n)2\pi t]\,dt$$

$$= 0 \quad \text{si} \quad m \neq n$$

$$= 1 \quad \text{si} \quad m = n$$

De manera similar :

$$\int 2\,\text{sen}(2\pi mt)\ \text{sen}(2\pi nt)\,dt\Big\} = 0 \text{ si } m \neq n$$

Pero para m = n ($0 \leq t \leq T$):

$$= \int 2 sen^2 (2\pi nt) dt = \int 2 \cos^2 (2\pi nt) dt = 1$$

Suponiendo que $S_n(t)$ fuera una función de tiempo y $S_m(t)$ la serie ortogonal, se puede observar que la Ec. 1.2 sólo tiene valor diferente de cero cuando se multiplica por la misma función, contenida tanto en f(t) como en la serie ortogonal. De esta manera, se puede descomponer una función f(t) en funciones seno o coseno; por ejemplo si se multiplica f(t) por una función que contenga al seno y al coseno como lo es $e^{-j\theta}$, se encontrarían las componentes senoidales que están en f(t).

Tomando en cuenta lo anterior la Transformada de Fourier se define como:

$$F(f) = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi ft}\, dt \qquad \text{Ec. 2.1}$$

donde: f(t) es una función en el dominio del tiempo.
$e^{-j2\pi ft} = \cos(2\pi ft) - j\, sen(2\pi ft)$ (forma Euler)
$j = \sqrt{-1}$
F(f) es una función en el dominio de la frecuencia.

La Transformada de Fourier está formada por una parte real y por una parte imaginaria, y cada una de ellas cumple con ciertas condiciones de simetría, por ejemplo, la Ec. 2.1 puede separarse en dos partes:

$$E(f) = \int_{-\infty}^{\infty} f(t) \cos(2\pi ft)\, dt \qquad \text{Ec. 2.2}$$

$$O(f) = \int_{-\infty}^{\infty} f(t)\, sen(2\pi ft)\, dt \qquad \text{Ec. 2.3}$$

donde E(f) corresponde a la parte real de la Transformada de Fourier[1] con la característica de tener simetría par[2] ("even", simetría con respecto al eje vertical, $f(t)=f(-t)$), y O(f), la que corresponde a la parte imaginaria, simetría non ("odd", $f(t)=-f(-t)$); de aquí F(f) se puede definir como:

$$F(f) = E(f) - j\,O(f) \qquad \text{Ec. 2.4}$$

$$|F(f)| = \sqrt{E(f)^2 + O(f)^2} \qquad \text{Ec. 2.5}$$

Ejemplos 2.1:

La Transformada de Fourier (figuras 2.2 a 2.4) de la función pulso $\Pi(t)$ (fig. 2.1):

$$\Pi(t) = \begin{cases} 0, & t < -½ \\ 1, & -½ < t < ½ \\ 0, & t > ½ \end{cases}$$

se puede expresar como:

$$F(f) = \int_{-\infty}^{\infty} \Pi\,(t - ½)\, e^{-2\pi f t}\, dt$$

$$F(f) = \int_0^1 e^{-2\pi f t}\, dt$$

$$= \frac{1}{2\pi f}\left[\operatorname{sen} 2\pi f t - j\cos 2\pi f t\right]_0^1$$

$$= \frac{1}{2\pi f}\left[\operatorname{sen} 2\pi f - j(1 - \cos 2\pi f)\right]$$

[1] Bracewell, 1965
[2] Ver ejemplo 2.1

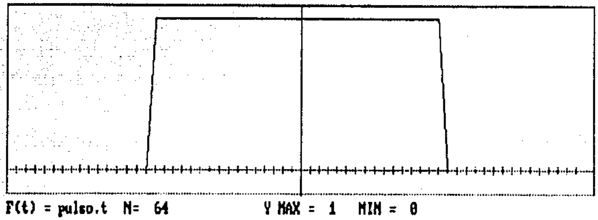


Figura 2.1 Función pulso .

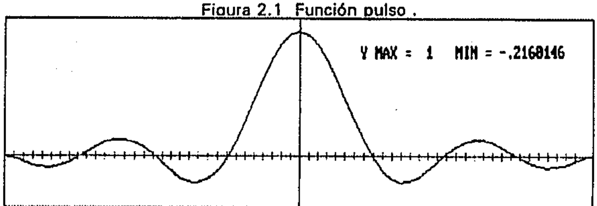


Figura 2.2 Transformada de Fourier (Parte Real)

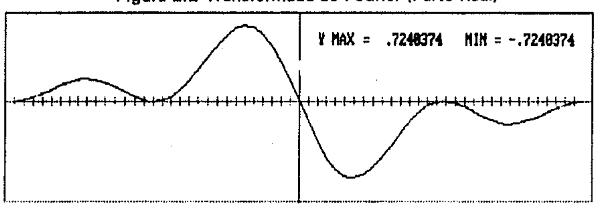


Figura 2.3 Transformada de Fourier (Parte Imaginaria)

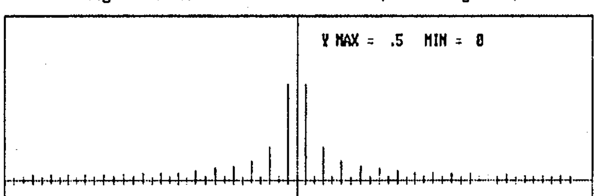


Figura 2.4 Transformada de Fourier (Magnitud).

Ejemplo 2.2:

La Transformada de Fourier (fig. 2.6) para la función $\cos(\omega_0 t)$ (figura 2.5) se expresa como:

$$F(f) = \lim_{\tau \to \infty} \int_{-\tau/2}^{\tau/2} \cos(\omega o t)\ e^{-jwt}\, dt$$

$$= \lim_{\tau \to \infty} \frac{\tau}{2} \left[\frac{\operatorname{sen}\left[(\omega - \omega_o) \frac{\tau}{2} \right]}{(\omega - \omega_o) \frac{\tau}{2}} + \frac{\operatorname{sen}\left[(\omega + \omega_o) \frac{\tau}{2} \right]}{(\omega + \omega_o) \frac{\tau}{2}} \right]$$

$$= \lim_{\tau \to \infty} \left[\frac{\tau}{2} \mathrm{Sa}\left[\frac{\tau(\omega - \omega_o)}{2} \right] + \frac{\tau}{2} \mathrm{Sa}\left[\frac{\tau(\omega + \omega_o)}{2} \right] \right]$$

donde si se considera la siguiente ecuación (Lathi, 1987):

$$\delta(t) = \lim_{\tau \to \infty} \frac{\tau}{\pi}\ \mathrm{Sa}(kt)$$

se tiene que:

$$F(f) = \pi\, [\, \delta(\omega - \omega o) + \delta(\omega + \omega o)\,]$$

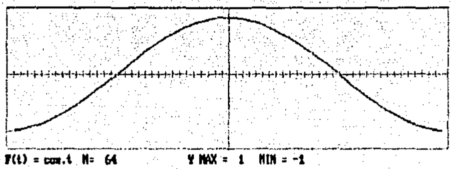


Figura 2.5 Función Cosenoidal

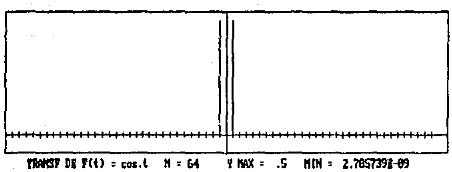


Figura 2.6 Transformada de Fourier[3]

2.2 PROPIEDADES DE LA TRANSFORMADA DE FOURIER.

Las propiedades de la Transformada de Fourier se pueden resumir en la siguientes tablas:

[3] Estas gráficas se obtuvierom via "Fast Fourier Transform".

Dominio del Tiempo f(t)	Dominio de la Frecuencia F(f)
Real	Parte real con simetría par Parte imaginaria con simetría impar
Imaginaria	Parte real con simetría impar Parte imaginaria con simetría par
Parte real con simetría par Parte imaginaria con simetría impar	Real
Parte real con simetría impar Parte imaginaria con simetría par	Imaginaria
Real y con simetría par	Real y con simetría par
Real y con simetría impar	Imaginaria y con simetría impar
Imaginaria y con simetría par	Imaginaria y con simetría par
Imaginaria y con simetría impar	Real y con simetría impar
Compleja y con simetría par	Compleja y con simetría par
Compleja y con simetría impar	Compleja y con simetría impar

Tabla 2.1 Características de la Transformada de Fourier dependiendo de la simetría de los datos.

En la tabla 2.1 se muestran las características que tiene la Transformada de Fourier dependiendo del tipo de simetría (non o par), o naturaleza (real o imaginaria) de los datos o señales de entrada. Algunas propiedades se pueden observar para los ejemplos 2.1 y 2.2 (real, real con simetría par).

Propiedad	Dominio del tiempo $f(t)$	Dominio de la frecuencia $F(f)$
Linealidad	$f_1(t) + f_2(t)$	$F_1(f) + F_2(f)$
Simetría	$f(t)$	$F(-f)$
Escalar	$f(kt)$	$k^{-1}F(f/k)$
Desplazamiento en el tiempo	$f(t - t_o)$	$F(f)e^{-j2\pi f t_o}$
Desplazamiento en la frecuencia	$f(t)e^{j2\pi t f_o}$	$F(f - f_o)$
Diferenciación en el tiempo $\omega = 2\pi f$	$\dfrac{d^n f(t)}{dt^n}$	$(j\omega)^n F(\omega)$
Diferenciación en la frecuencia	$(-jt)^n f(t)$	$\dfrac{d^n F(\omega)}{d\omega^n}$
Integración en el tiempo	$\int_{-\infty}^{t} f(\tau)\, d\tau$	$(j\omega)^{-1}F(\omega)$
Convolución en el tiempo	$f_1(t) * f_2(t)$ Ec. 2.7	$F_1(\omega)F_2(\omega)$ Ec. 2.8
Convolución en la frecuencia	$f_1(t)f_2(t)$ Ec. 2.9	$(2\pi)^{-1}[F_1(\omega) * F_2(\omega)]$

Tabla 2.2 Propiedades de la Transformada de Fourier.

2.3 TRANSFORMADA INVERSA DE FOURIER.

La transformada inversa de Fourier se define como (Brigham, 1974):

$$f(t) = \int_{-\infty}^{\infty} F(f)\, e^{j2\pi f t}\, df \qquad \text{Ec. 2.6}$$

donde: $F(f)$ es una función en el dominio de la frecuencia.
$f(t)$ es una función en el dominio del tiempo.

Esta ecuación permite encontrar la función en tiempo de una Transformada de Fourier. La Transformada y su antitransformada , es decir, la función en el dominio de la frecuencia y del tiempo, pueden ser expresadas mediante la siguiente notación:

$$f(t) \Leftrightarrow F(f)$$

A esta relación se le conoce como **par de Transformadas de Fourier.**

2.4 INTEGRAL DE CONVOLUCIÓN

La integral de convolución está dada por:

$$y(t) = \int_{-\infty}^{\infty} x(\tau)\, h(t-\tau)\, d\tau = x(t) * h(t) \qquad \text{Ec. 2.7}$$

donde el teorema de convolución se define como:

$$h(t) * x(t) \Leftrightarrow H(f)\; X(f) \qquad \text{Ec 2.8}$$

Esto significa que la convolución en el tiempo de dos funciones es igual a su multiplicación en el dominio de la frecuencia.

A su vez el teorema de convolución en la frecuencia se puede escribir:

$$h(t)\, x(t) \Leftrightarrow H(f) * X(f) \qquad \text{Ec 2.9}$$

2.5 SERIES DE FOURIER

La serie de Fourier se puede obtener como un caso particular de la Integral de Fourier (aunque también se puede obtener de manera independiente[4]).

La serie de Fourier para una función periódica f(t) se representa:

$$f(t) = 0.5a_0 + \sum_{n=1}^{\infty}\left[a_n \cos(2\pi f_0 nt) + b_n \mathrm{sen}(2\pi f_0 nt)\right] \qquad \text{Ec. 2.10}$$

donde: f_o es la frecuencia fundamental (1/T)
T es el período

Cada uno de los coeficientes a_n y b_n se pueden obtener del siguiente par de integrales:

$$a_n = \int_{-T/2}^{T/2} f(t) \cos(2\pi fnt)\, dt \qquad n = 0,1,2,3...... \qquad \text{Ec. 2.11}$$

$$b_n = \int_{-T/2}^{T/2} f(t) \mathrm{sen}(2\pi fnt)\, dt \qquad n = 0,1,2,3.... \qquad \text{Ec. 2.12}$$

Para derivar esta serie de la integral de Fourier, la función periódica f(t) se representa mediante la convolución en el tiempo de dos funciones, h(t) y x(t), donde h(t) representa sólo un periodo de f(t) y x(t) es una serie de impulsos que que se encuentran equidistantes T unidades de tiempo, donde T es el periodo de la función f(t).

---

[4] Ver capítulo II

2.6 TRANSFORMADA DISCRETA DE FOURIER (DFT[5]).

En la figura 2.7 se puede observar la Transformada Discreta de Fourier, que se aplica a una señal muestreada en el tiempo. Debido a la "periodicidad" tanto en el dominio del tiempo como el de la frecuencia, sólo un número finito de muestras se requieren para la transformada y esto permite que sea calculada directamente por medios digitales o numéricos. De hecho, si un periodo de una señal en el tiempo puede ser descrito por N muestras, el espectro de frecuencia también va a contener N muestras por periodo. Sin embargo, debido a la simetría del espectro de frecuencia (que se presenta en las funciones o señales reales en el tiempo) sólo se requieren N/2 muestras[6].

La Transformada Discreta puede ser derivada de la Integral de Fourier utilizando los siguientes tres pasos: (fig. 2.7)

1) Muestreo en el Tiempo

La señal continua y su espectro de frecuencia obtenido usando la Integral de Fourier, se muestra en la figura 2.7a. La señal en el tiempo es muestreada multiplicándola por la función de muestreo. Esta función consiste en una serie infinita de impulsos con una separación de Δt. La transformada de Fourier de la señal muestreada, es otra serie de impulsos con una separación $f_s = (1/\Delta t)$. De acuerdo con el Teorema de la Convolución, la multiplicación de dos señales en el tiempo equivale a la convolución de cada uno de sus espectros. Los resultados se muestran en la figura c donde se puede observar la periodicidad del espectro de frecuencias que corresponde a la señal muestreada. De aquí que si una señal se muestrea por debajo del criterio de Nyquist el espectro constará de repeticiones de este mismo que se traslapan. A este error se le denomina doblez espectral (aliasing) o error de pseudo interferencia (Lathi, 1986).

[5] Discrete Fourier Transform
[6] Ver tabla 2.1

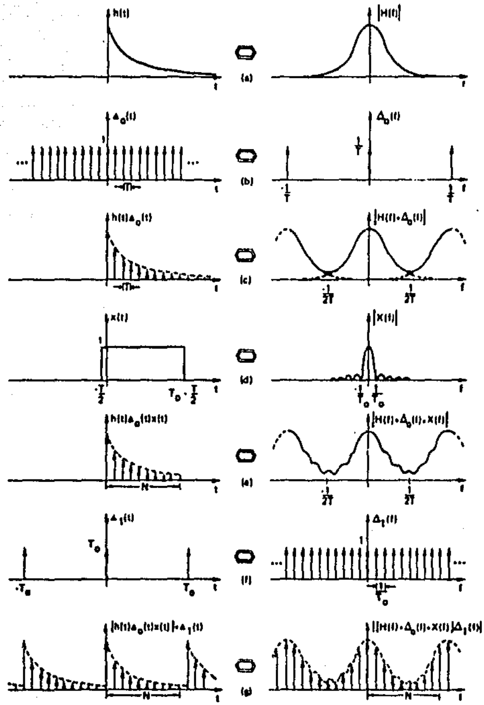


Fig. 2.7 Derivación de la Transformada Discreta de Fourier.

2) Limitar la señal en el Tiempo.

Para limitar el número de muestras en la función, la señal del tiempo muestreada se multiplica por una función de ventana. Esto se muestra en la figura d por medio de una señal rectangular de longitud T. La transformada de Fourier de esta función es la función[7] Sa(x) mostrada en la figura d. El resultado de la multiplicación por la función de ventana se observa en la figura e donde el número de muestras $N = T/\Delta t$. En el dominio de la frecuencia la multiplicación por la ventana de tiempo se puede representar como la convolución de sus transformadas (Ec. 2.6) en la frecuencia, lo que introduce "rizos" y lóbulos laterales ("ripples, sidelobes, leakage") en el espectro, debido a la forma de la función Sa(x). Este efecto se reduce si utilizamos ventanas como la de Hanning (figura 2.8), cuya función es:

$$H(t) = 0.5[1-\cos(2\pi t/T_c)] \qquad \text{Ec. 2.13}$$

donde: $0 \le t \le T_c$

T_c es el periodo, intervalo de truncamiento o tamaño de la ventana:

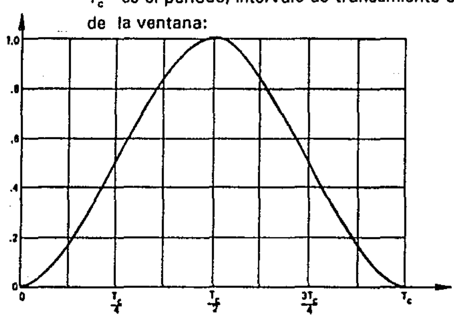


Figura 2.8 Ventana de Hanning

[7] $Sa(x) = \operatorname{sen} x/x$

Otra forma de reducir los efectos de rizo es incrementando el periodo de la función de ventana; con esto se puede observar que incrementando la longitud en el tiempo (propiedad escalar, tabla 2.2) se reduce el ancho de banda de la señal en frecuencia. (figura 2.7d)

3) Convolución en el tiempo.

Si se calcula computacionalmente la transformada sólo se pueden utilizar valores muestreados del espectro de frecuencia (figura 2.7e), por lo que para discretizar la transformada es necesario modificarla, multiplicándola por una función de muestreo (tren de impulsos) con intervalos de $\Delta f = 1/T$ (figura 2.7f) correspondientes a N ($N = T/\Delta t = Tf_s$) muestras dentro del periodo del dominio de la frecuencia. En el dominio del tiempo, el muestreo en la frecuencia corresponde a la convolución de la señal limitada en tiempo con impulsos de separación T, lo cual produce una señal de tiempo periódica cuyo periodo es la señal de tiempo limitada. Este resultado es la Transformada Discreta de Fourier (DFT) (figura 2.7g).

La Transformada Discreta y su antitransformada están dadas por el siguiente par de ecuaciones (Montaño, 1989):

$$F(k) = 1/N \sum_{n=0}^{N-1} f(nT)\, e^{-2\pi jnk/N} \qquad \text{Ec. } 2.14$$

$$f(nT) = \sum_{n=0}^{N-1} F(k)\, e^{2\pi jnk/N} \qquad \text{Ec. } 2.15$$

donde: T es el período de muestreo
k es la variable discreta en el dominio de la frecuencia
N es el número total de puntos

Propiedad	Dominio del tiempo $f(nT)$	Dominio de la frecuencia $F(k)$
Linealidad	$f1(nT) + f2(nT)$	$F1(k) + F2(k)$
Simetría	$(1/N)\ f(nT)$	$F(-n)$
Desplazamiento en el tiempo	$f(nT-i)$	$F(k)e^{-j2\pi ni/N}$
Desplazamiento en la frecuencia	$f(nT)e^{j2\pi ki/N}$	$F(n-i)$
Convolución en el tiempo	$f_1(nT) * f_2(nT)$	$F_1(k)F_2(k)$
Convolución en la frecuencia	$f_1(nT)f_2(nT)$	$N^{-1}[F_1(k) * F_2(k)]$

Tabla 2.3
Propiedades de la Transformada Discreta de Fourier

2.7 TRANSFORMADA RÁPIDA DE FOURIER (FFT).

El algoritmo utilizado para la Transformada Rápida fue el de Sande-Tukey (Montaño, 1989) y fue programado en QuickBasic. La Transformada Rápida de Fourier reduce de N^2 multiplicaciones que se realizan en la transformada discreta a $N\log_2N$ operaciones, donde N es el número de datos. Los datos de entrada estan en el vector F(f) y el resultado de la transformada también se colocan en el mismo vector para "economizar" memoria. El programa implementado como una subrutina se lista a continuación:

```
        'SUBRUTINA PARA CALCULAR LA TRANSFORMADA  RAPIDA DE FOURIER

   AG# = PI * 2 / N
   FOR Z = 0 TO N / 2 - 1
     WRE#(Z) = COS(AG# * (-Z))
     WIM#(Z) = -SIN(AG# * (-Z))
   NEXT Z
**************************************************************
   'CALCULO DE LA FFT
**************************************************************
```

```
    FOR I = 1 TO P
        L = 0: H = 0: G = N / (2 ^ I)
        FOR k = 0 TO N - 1 STEP G
          TFI = 0:TFIFLAG = (-1) ^ (L + 1)
          FOR J = 0 TO G - 1
            TFI = J * 2 ^ (I - 1)
            TFI = 0:S = J + H:T = J + G + H
            IF TFIFLAG > 0 THEN
                  TEMPRE = F(S) - F(T)
                  TEMPIM = I(S) - I(T)
                  BUFERE(R) = TEMPRE * WRE#(TFI) - TEMPIM * WIM#(TFI)
                  BUFEIM(R) = TEMPRE * WIM#(TFI) + TEMPIM * WRE#(TFI)
            ELSE
                  BUFERE(R) = F(S) + F(T)
                  BUFEIM(R) = I(S) + I(T)
            END IF
          NEXT J
          L = L + 1:H = INT(L / 2) * G * 2
        NEXT k
        FOR II = 0 TO N - 1
          F(II) = BUFERE(II): I(II) = BUFEIM(II)
        NEXT II
    NEXT I
    FOR I = 0 TO N - 1
        F(I) = F(I) / N:I(I) = I(I) / N
    NEXT I
'*************************************
  ' REORDENAMIENTO DE BITS
'*************************************
    FOR I = 0 TO N - 1
        INDEX% = I:IOUT% = 0
        FOR J = 1 TO P
          TEMP% = 1 AND INDEX%: IOUT% = IOUT% * 2: IOUT% = IOUT% + TEMP%
          INDEX% = INDEX% \ 2
        NEXT J
        BUFERE(I) = F(IOUT%):BUFEIM(I) = I(IOUT%)
    NEXT I
'*******************************************
    'CALCULO DE MODULOS Y FASES
'*******************************************
    FOR I = 0 TO N - 1
      BUFERE(N) = BUFERE(0): BUFEIM(N) = BUFEIM(0)
      AV(I) = (BUFERE(I) + BUFERE(N - I)) / 2: BV(I) = (BUFEIM(I) - BUFEIM(N - I)) / 2
    NEXT I
    F(0) = AV(0)
    FOR J = 1 TO N - 1
      F(J) = SQR((BV(J) ^ 2 + AV(J) ^ 2)): 'Modulo j
      IF AV(J) = 0 THEN
          IF BV(J) < 0 THEN FASE(J) = PI / 2: IF BV(J) > 0 THEN FASE(J) = 3 * PI / 2
      END IF
      IF AV(J) < > 0 THEN FASE(J) = ATN(-BV(J) / AV(J))
      FASE(J) = 360 * FASE(J) / 2 / PI

    NEXT J

RETURN
```

CAPITULO III .- TRANSFORMADA DE HARTLEY

3.1 INTRODUCCION.

Otra transformada utilizada es la Transformada de Hartley (TH). Esta herramienta matemática apareció por primera vez en el "Proceedings of Radio Engineers" en el año de 1942 y fue desarrollada por Ralph V.L. Hartley. La TH no utiliza números imaginarios y se puede obtener la misma información de amplitud y fase que la que se obtiene del espectro de Fourier.

3.2 ANTECEDENTES.

Hartley (1890-1970) trabajaba en los Laboratorios de investigación de la compañía Western Electric, dirigiendo los primeros trabajos de desarrollo de receptores de radio destinados a un radio teléfono trasatlántico. Por esta época inventa el circuito oscilador que lleva su nombre (Cooper, 1982).

Durante la primera guerra mundial, investigó la forma en que el oyente, por medio de mecanismos auditivos y cerebrales, percibe la dirección de donde se genera un sonido. Después de la guerra, trabajó en los laboratorios Bell donde formuló un importante principio de la tecnología de información que enuncia que la cantidad total de información que un sistema puede transmitir, es proporcional al producto de la banda de frecuencias que el sistema transmite por el tiempo, durante el cual el sistema está disponible para transmitir.

En 1929, renuncia a la dirección de su grupo por motivos de salud, pero cuando se mejora, se dedica a los estudios teóricos de lo que hoy conocemos como **"Transformada de Hartley"**.

3.3 DEFINICION. TRANSFORMADA DE HARTLEY.

Considerando el mismo desarrollo aplicado en el capítulo 2 para la Transformada de Fourier, la Transformada de Hartley se define como:

$$H(f) = \int_{-\infty}^{\infty} f(t)\ \mathrm{cas}\ (2\pi ft)\ dt \qquad \text{Ec. 3.1}$$

donde: $\mathrm{cas}\ (2\pi f t) = \cos(2\pi f t) + \mathrm{sen}\ (2\pi f t)$
$f(t)$ es una función en el dominio del tiempo
$H(f)$ es un a función en el dominio de la frecuencia.

Como la única diferencia en cuanto a las definiciones de la Transformada de Fourier (Ec. 2.1) y la Transformada de Hartley (Ec. 3.1) es el valor de j ($\sqrt{-1}$) las propiedades de simetría son las mismas, por lo que la Ec. 3.1 se puede escribir en función de sus componentes de simetría par y non (Ecs. 2.2 y 2.3) como:

$$H(f) = E(f) + O(f) \qquad \text{Ec. 3.2}$$

Ejemplo 3.1:

La Transformada de Hartley de la función pulso $\Pi(t)$ (Figs. 3.1 y 3.2)

$$\Pi(t) = \begin{cases} 0, & t < -\frac{1}{2} \\ 1, & -\frac{1}{2} < t < \frac{1}{2} \\ 0, & t > \frac{1}{2} \end{cases}$$

se expresa como:

$$H(f) = \int_{-\infty}^{\infty} \Pi\ (t - \tfrac{1}{2})\ \mathrm{cas}(2\pi ft)\ dt$$

$$H(f) = \int_0^1 \text{cas}\,(2\pi ft)\,dt$$

$$= \frac{1}{2\pi f}\left[\text{sen}2\pi ft - \cos 2\pi ft\right]_0^1$$

$$= \frac{1}{2\pi f}\left[\text{sen}2\pi f - \cos 2\pi f + 1\right]$$

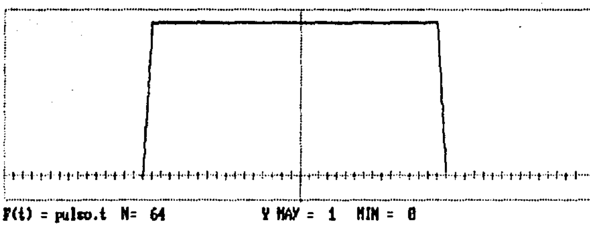


Fig. 3.1 Función pulso

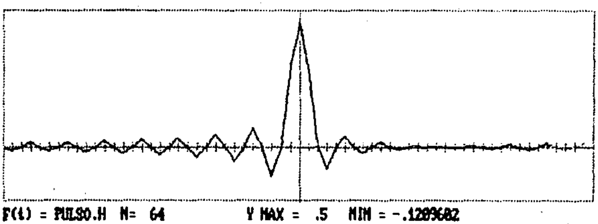


Fig 3.2 Transformada de Hartley de la función pulso

Ejemplo 3.2:

La Transformada de Hartley de la función cos (ωot) (Figs 3.3 y 3.4) se puede obtener como:

$$H(f) = \lim_{\tau \to \infty} \int_{-\tau/2}^{\tau/2} \cos(\omega o t) \text{ cas } wt \, dt$$

$$= \lim_{\tau\to\infty} \frac{\tau}{2}\left[\frac{\text{sen}\left[(\omega-\omega_o)\frac{\tau}{2}\right]}{(\omega-\omega_o)\frac{\tau}{2}} + \frac{\text{sen}\left[(\omega+\omega_o)\frac{\tau}{2}\right]}{(\omega+\omega_o)\frac{\tau}{2}}\right]$$

$$= \lim_{\tau\to\infty} \left[\frac{\tau}{2}\text{Sa}\left[\frac{\tau(\omega-\omega_o)}{2}\right] + \frac{\tau}{2}\text{Sa}\left[\frac{\tau(\omega+\omega_o)}{2}\right]\right]$$

donde si se considera la siguiente ecuación (Lathi, 1987):

$$\delta(t) = \lim_{\tau \to \infty} \frac{\tau}{\pi} \text{ Sa}(kt)$$

se tiene que:

$$H(f) = \pi [\, \delta(\omega - \omega o) + \delta(\omega + \omega o)\,]$$

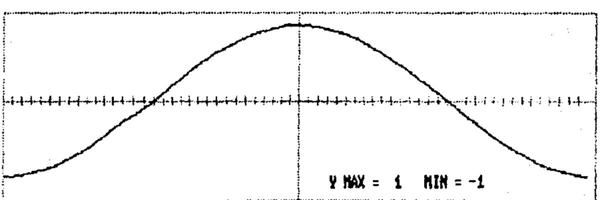
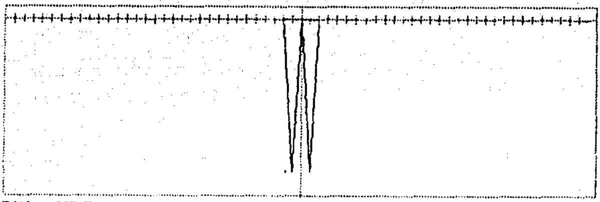


Fig. 3.3 Función $\cos(\omega_o t)$



F(t) = COS.H N= 64 Y MAX = 6.608409E-08 MIN = -.5

Fig. 3.4 Transformada de Hartley

3.4 RELACION ENTRE LA TRANSFORMADA DE HARTLEY Y LA TRANSFORMADA DE FOURIER.

Considerando las propiedades de simetría y los ejemplos analizados (ejs. 2.1, 2.2, 3.1, 3.2) la Transformada de Fourier se puede obtener por medio de la transformada de Hartley de la siguiente manera:

De la ec. 3.2 se puede expresar las partes non y par en función de la Transformada de Hartley como:

$$E(f) = H(f) + H(-f)$$

$$O(f) = H(f) - H(-f) \qquad \text{Ecs. 3.3}$$

donde: **E(f) y O(f) corresponden a la parte real e imaginaria de la Transformada de fourier.**

Sustituyendo en la Ec. 2.5 se tiene que:

$$P(f) = E(f)^2 + O(f)^2$$

$$= |H(f) + H(-f)|^2 + |H(f) - H(-f)|^2$$

$$= \frac{[H(f)]^2 + [H(-f)]^2}{2} \qquad \text{Ec. 3.4}$$

Haciendo un análisis similar para la fase:

$$\text{Fase (f)} = \text{ang tan}\left[-\frac{O(f)}{E(f)}\right]$$

$$= \text{ang tan}\left[-\frac{H(f)-H(-f)}{H(f)+H(-f)}\right] \qquad \text{Ec. 3.5}$$

De las ecs. 3.4 y 3.5 se puede observar que se obtiene la misma información que la Transformada de Fourier , tanto de Magnitud como de Fase (Bracewell, 1989). Así para el ejemplo 3.1 y 3.2 el espectro por medio de la Transformada de Hartley es:

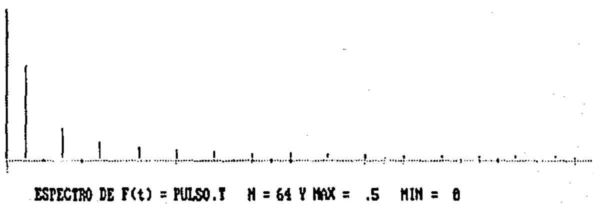


Fig. 3.5 Espectro de la Función Pulso

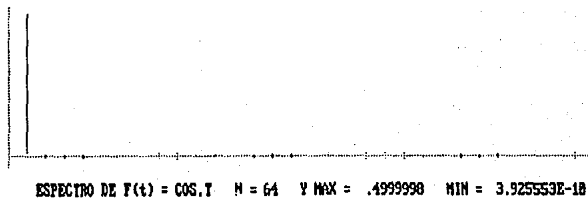


Fig. 3.6 Espectro de la función coseno

Como se puede observar se obtienen los mismos resultados que las figs. 2.4 y 2.6.

3.5 PROPIEDADES DE LA TRANSFORMADA DE HARTLEY.

PROPIEDAD	DOMINIO DEL TIEMPO $f(t)$	DOMINIO DE LA FRECUENCIA $F(f)$
Linealidad	$f_1(t) + f_2(t)$	$H_1(f) + H_2(f)$
Escalar	$f(kt)$	$k^{-1} H(f/k)$
Desplazamiento en el tiempo	$f(t - t_0)$	$H(-f)\operatorname{sen}2\pi f t_0 + H(f)\cos 2\pi f t_0$
Diferenciación en el tiempo	$\frac{d^n f(t)}{dt^n}$	$-2\pi f\, H(-f)$
Convolución en el tiempo.	$f_1(t) * f_2(t)$	$\frac{1}{2}[H_1(f)H_2(f) - H_1(-f)H_2(-f) + H_1(f)H_2(-f) + H_1(-f)H_2(-f)]$
Convolución en la frecuencia	$f_1(t)f_2(t)$	$\frac{1}{2}[H_1(f)*H_2(f) - H_1(-f)*H_2(-f) + H_1(f)*H_2(-f) + H_1(-f)*H_2(-f)]$

Tabla 3.1

PROPIEDADES DE LA FUNCION CAS(t)	
$\operatorname{cas}(-t) = \operatorname{cas}'(t)$	$\cos t - \operatorname{sen} t$
$\operatorname{cas}(A+B)$	$\cos A \operatorname{cas} B + \operatorname{sen} A \operatorname{cas}' B$
$\operatorname{cas}(A-B)$	$\cos A \operatorname{cas}' B + \operatorname{sen} A \operatorname{cas} B$
$\operatorname{cas} A \operatorname{cas} B$	$\cos(A-B) + \operatorname{sen}(A+B)$
$\operatorname{cas} A + \operatorname{cas} B$	$2 \operatorname{cas} \frac{1}{2}(A+B) \cos \frac{1}{2}(A-B)$
$\operatorname{cas} A - \operatorname{cas} B$	$2 \operatorname{cas}' \frac{1}{2}(A+B) \operatorname{sen} \frac{1}{2}(A-B)$

Tabla 3.2

3.6 TRANSFORMADA INVERSA

La Transformada inversa de Hartley se puede definir como:

$$f(t) = \int_{-\infty}^{\infty} H(f) \text{ cas } (2\pi ft) \, dt \qquad \text{Ec. 3.6}$$

donde: $f(t)$ es la función en el tiempo
$H(f)$ es la transformada de Hartely de $f(t)$

Como se puede observar la Transformada (Ec. 3.1) y su inversa utilizan el mismo término "cas t", por lo que se utiliza el mismo algoritmo para los dos casos[1].

3.7 TRANSFORMADA DISCRETA.(DHT[2])

Para deducir la Transformada Discreta de Hartley se puede hacer un análisis similar al de Fourier (sección 2.6). De esta manera la transformada discreta de Hartley y su antitransformada están dadas por :

$$H(\nu) = N^{-1} \sum_{\tau=0}^{N-1} f(\tau) \text{ cas } (2\pi\nu\tau / N) \qquad \text{Ec.3.7}$$

$$f(\tau) = \sum_{\nu=0}^{N-1} H(\nu) \text{ cas } (2\pi\nu\tau / N) \qquad \text{Ec.3.8}$$

---

[1] Esta característica en el momento de implementarse computacionalmente equivale a la mitad de los recursos en cuanto a programa (memoria)

[2] Discrete Hartley Transformation.

PROPIEDADES	DOMINIO DEL TIEMPO $f(\tau)$	DOMINIO DE LA FRECUENCIA $F(\upsilon)$
Linealidad	$f_1(\tau) + f_2(\tau)$	$H_1(\upsilon) + H_2(\upsilon)$
Desplazamiento en el tiempo	$f(\tau - T)$	$-H(N-\upsilon)sen(2\pi T\upsilon/N) + H(\upsilon)cos(2\pi\upsilon T/N)$
Derivación en tiempo	$\frac{d^n f(\tau)}{dt^n}$	$2\pi\upsilon H(-\upsilon)$
Convolución en el tiempo.	$f_1(\tau)*f_2(\tau)$	$½N\,[H_1(\upsilon)H_2(\upsilon)-H_1(-\upsilon)H_2(-\upsilon) + H_1(\upsilon)H_2(-\upsilon) + H_1(-\upsilon)H_2(-\upsilon)]$
Convolución en la frecuencia	$f_1(\tau)f_2(\tau)$	$½\,N\,[H_1(\upsilon)*H_2(\upsilon) -H_1(-\upsilon)*H_2(-\upsilon) + H_1(\upsilon)*H_2(-\upsilon) + H_1(-\upsilon)*H_2(-\upsilon)]$

Tabla 3.3 Propiedades de la Transformada Discreta de Hartley

3.8 TRANSFORMADA RAPIDA DE HARTLEY (FHT[3]).

El algoritmo utilizado para la Transformada Rápida (FHT) es el de Bracewell[4] (Bracewell,1986), este programa se implantó como una subrutina. El vector H(f) corresponde a los datos de entrada, y para "economizar" memoria se utilizó el mismo vector para los datos de salida. Al igual que la Transformada Rápida de Fourier el número de operaciones se reduce de N^2 a $N\log_2 N$.

HARTLEY:

```
'*******************************************************
'                FHTSUB
' Esta subrutina toma los datos en el vector F(f)
' y calcula al Transformada en el mismo vector
'*******************************************************

'**************REACOMODAR DATOS PARA LA TRANSFORMADA
GOSUB READAF
T0 = TIMER:    Tiempo de c lculo
CLS
```

[3] Fast Hartley Transformation

[4] Existen otros algoritmos ver Narayanan & Prablu, 1991

```
9030 IF P = 1 THEN
        J = f(0) + f(1)
        f(1) = f(0) - f(1)
        f(0) = J
        RETURN
    END IF
    N9 = 2 ^ (P - 2)
    NP = 4 * N9
    C9(5) = NP - 1
    C9(6) = P - 1
    IF NP = N0 THEN GOTO 9400              'Salta pretabulacion
    I = 1
    M9(0) = 1
    M9(1) = 2

9202  M9(I + 1) = M9(I) + M9(I)
    I = I + 1
    IF I < P THEN GOTO 9202
    IF NP = 2 THEN GOTO 9411              'Caso especial
    IF NP < 8 THEN GOTO 9400              'Salta funciones trigonometricas
    S9(N9) = 1
    IF NP = 8 THEN S9(1) = SIN(pi / 4): GOTO 9330                  'Salta senos

    '**************************************************
9300 'OBTENCION DEL SENO
    '**************************************************
    FOR I = 1 TO 3
        S9(I * N9 / 4) = SIN(I * pi / 8)
    NEXT I
    H9 = 1 / 2 / COS(pi / 16)        'INICIO DE SECANTE

    C9(4) = P - 4
    FOR I = 1 TO (P - 4)
        C9(4) = C9(4) - 1
        V9(0) = 0
        FOR J = M9(C9(4)) TO (N9 - M9(C9(4))) STEP M9(C9(4) + 1)
         V9(1) = J + M9(C9(4))
         S9(J) = H9 * (S9(V9(1)) + V9(0))
         V9(0) = S9(V9(1))
        NEXT J
        H9 = 1 / SQR(2 + 1 / H9):
        NEXT I

9330 '*******OBTENCION DE TANGENTES*******
    C9(0) = N9 - 1
    FOR I = 1 TO (N9 - 1)
        T9(I) = (1 - S9(C9(0))) / S9(I)
        C9(0) = C9(0) - 1
    NEXT I
    T9(N9) = 1

9400 '****************************************************************
    'PERMUTACION RAPIDA
```

```
'*****************************************************************
'*****PARA  P = 2, 3 permutacion directa*****
IF P = 2 THEN
      V9(9) = f(1)
      f(1) = f(2)
      f(2) = V9(9)
      GOTO 9500
END IF
IF P = 3 THEN
      V9(9) = f(1)
      f(1) = f(4)
      f(4) = V9(9)
      V9(9) = f(3)
      f(3) = f(6)
      f(6) = V9(9)
END IF
IF P = 3 THEN GOTO 9500

'******* Para P = 4,5,6 (Q9 = 2,3 ), salta la tabla*****
Q9 = INT(P / 2)
C9(2) = M9(Q9)
Q9 = Q9 + P MOD 2
IF Q9 = 2 THEN
      A9(1) = 2
      A9(2) = 1
      A9(3) = 3
      GOTO 9420
END IF
IF Q9 = 3 THEN
 A9(1) = 4
 A9(2) = 2
 A9(3) = 6
 A9(4) = 1
 A9(5) = 5
 A9(6) = 3
 A9(7) = 7
 GOTO 9420
END IF

9411 IF NP = 2 THEN
 V9(6) = f(0)
 f(0) = f(1)
 f(1) = V9(6)   'caso especial
END IF

A9(0) = 0
A9(1) = 1
FOR I = 2 TO Q9
 FOR J = 0 TO (M9(I - 1) - 1)
      A9(J) = A9(J) + A9(J)
      A9(J + M9(I - 1)) = A9(J) + 1
 NEXT J
NEXT I
```

```
9420 FOR I = 1 TO (C9(2) - 1)
      V9(4) = C9(2) * A9(I)
      V9(5) = I
      V9(6) = V9(4)
      V9(7) = f(V9(5))
      f(V9(5)) = f(V9(6))
      f(V9(6)) = V9(7)
      FOR J = 1 TO (A9(I) - 1)
            V9(5) = V9(5) + C9(2)
            V9(6) = V9(4) + A9(J)
            V9(7) = f(V9(5))
            f(V9(5)) = f(V9(6))
            f(V9(6)) = V9(7)
      NEXT J
    NEXT I

9500 '******    estados 1 & 2         ******
    '******  obt,n dos elementos DHTs  ******
    FOR I = 0 TO (NP - 2) STEP 2
      V9(6) = f(I) + f(I + 1)
      V9(7) = f(I) - f(I + 1)
      f(I) = V9(6)
      f(I + 1) = V9(7)
    NEXT I
    IF P = 1 THEN RETURN

9510 '****** obtencion de 4 elementos *****
    FOR I = 0 TO (N - 4) STEP 4
      V9(6) = f(I) + f(I + 2)
      V9(7) = f(I + 1) + f(I + 3)
      V9(8) = f(I) - f(I + 2)
      V9(9) = f(I + 1) - f(I + 3)
      f(I) = V9(6)
      f(I + 1) = V9(7)
      f(I + 2) = V9(8)
      f(I + 3) = V9(9)
    NEXT I

9520 IF P = 2 THEN RETURN:

9600 '******* estados 3 & 4 **********
    U9 = C9(6)
    S9 = 4
    FOR L9 = 2 TO C9(6)
      V9(2) = S9 + S9
      U9 = U9 - 1
      V9(3) = M9(U9 - 1)
      FOR Q9 = 0 TO C9(5) STEP V9(2)
            I = Q9
            D9 = I + S9
            V9(6) = f(I) + f(D9)
            V9(7) = f(I) - f(D9)
            f(I) = V9(6)
            f(D9) = V9(7)
```

```
            K9 = D9 - 1
            FOR J = V9(3) TO N9 STEP V9(3)
             I = I + 1
             D9 = I + S9
             E9 = K9 + S9
             V9(9) = f(D9) + f(E9) * T9(J)
             X9 = f(E9) - V9(9) * S9(J)
             Y9 = X9 * T9(J) + V9(9)
             V9(6) = f(I) + Y9
             V9(7) = f(I) - Y9
             V9(8) = f(K9) - X9
             V9(9) = f(K9) + X9
             f(I) = V9(6)
             f(D9) = V9(7)
             f(K9) = V9(8)
             f(E9) = V9(9)
             K9 = K9 - 1
            NEXT J
            E9 = K9 + S9
      NEXT Q9
      S9 = V9(2)
    NEXT L9
    N0 = NP
    DURA = TIMER - T0
    FOR I = 0 TO N - 1
          f(I) = f(I) / N
    NEXT I
    PRINT "TIEMPO = ", DURA
    PRINT "Presione cualquier tecla"

LB9: A$ = INKEY$: IF A$ = "" THEN GOTO LB9
    '*************//REORDENAR DATOS
    GOSUB READAF
RETURN
```

CAPITULO IV .- TRANSFORMADA DE WALSH

4.1 INTRODUCCIÓN

Continuando con la idea de descomponer una función en un conjunto de funciones ortogonales, se tienen las funciones de Walsh, las cuales son un conjunto de funciones rectangulares con valores de 1 y -1 únicamente.

Este tipo de funciones por su naturaleza permiten descomponer funciones con discontinuidades, o funciones de tipo impulso o escalón.

4.2 ANTECEDENTES

Las funciones de Walsh fueron definidas en 1923 por el matemático americano J. L. Walsh. En el artículo original, Walsh dio una definición recursiva de sus funciones de acuerdo al número promedio de "cruces" por cero para la función en un intervalo de tiempo. Este ordenamiento, también lo usó el matemático Polaco S. Karzmarz en sus trabajos sobre series, el cual, será utilizado más adelante como orden de Walsh-Kacmarz.

Tiempo después H. F. Harmuth, propone el término de "orden secuencial" para este ordenamiento, el cual también es bastante utilizado.

En 1931 R.E.A. L. Paley, otro matemático americano, dá una definición diferente de las funciones de Walsh, y las define a partir de productos de funciones de Rademacher obteniendo un orden completamente distinto del mencionado anteriormente. A este orden se le conoce como "orden natural" u "ordenamiento binario". La relación entre estos órdenes fue dada por F. Pichler.

Otra aproximación a las funciones de Walsh está dado por matrices ortogonales, cuyos elementos sólo tienen valores de 1 y -1.

Estas matrices fueron trabajadas por el matemático británico J. J. Sylvester en 1867, y generalizadas por el matemático francés M.J. Hadamard en 1893, quién establece este tipo de matrices, conocidas actualmente como matrices de Hadamard. Las funciones de Walsh obtenidas por este tipo de matrices tienen un orden diferente llamado ordenamiento de Kronecker.

4.3 FUNCIONES DE WALSH

Las funciones de Walsh forman un conjunto ordenado de funciones rectangulares, y sólo toman dos valores: 1 y -1. Estas funciones se definen en un período de tiempo T, conocido como base de tiempo que puede ser de [0,1] o de [-½, ½].

Para definir cada una de las funciones de Walsh se requiere de dos términos: Un término n que indica el *"orden"*, relacionado con su frecuencia, que para este caso es el número de cruces por cero (orden secuencial) y un tiempo t, normalizado con respecto a la base de tiempo t/T. De esta manera la función se puede describir como:

$$WAL(n, t)$$

Las funciones de Walsh por ser un conjunto ortogonal de funciones deben cumplir que:

$$\sum_{t=0}^{N-1} WAL(m,t)\, WAL(n,t)\, dt = \begin{cases} N & \text{si } n = m \\ 0 & \text{si } n \neq m \end{cases} \qquad \text{Ec. 4.1}$$

Las primeras 32 funciones de Walsh[1] en orden secuencial o orden de Walsh-Kacmarz con fase positiva[2] se muestran en la figura 4.1x.

[1] Estas funciones se obtuvieron de la ec. 4.10. y con el programa para obtener funciones de Walsh, ver apéndice.
[2] Fase positiva significa que la base de tiempo está en el intervalo [0,1]

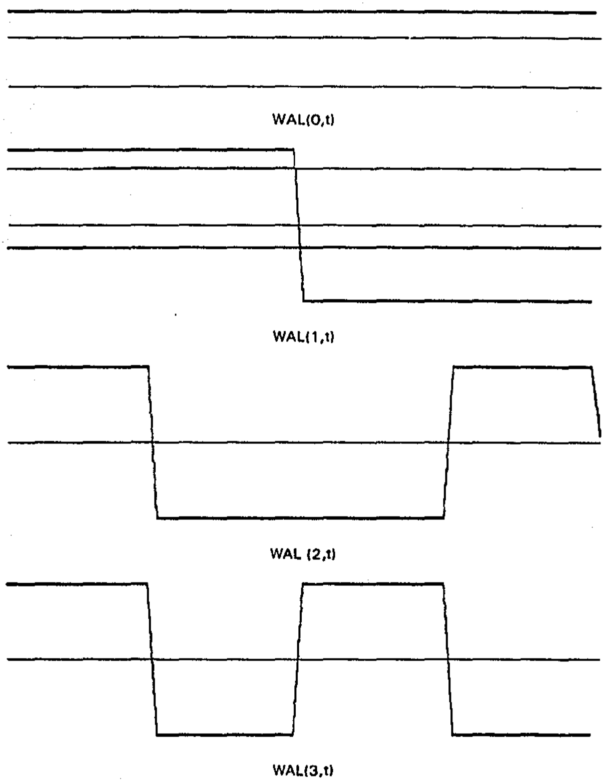
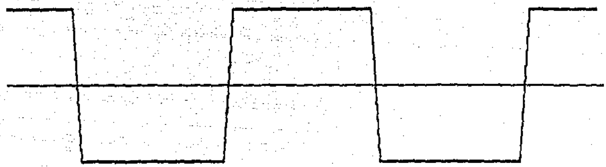
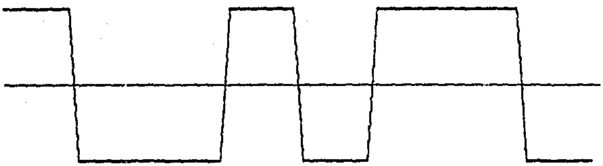


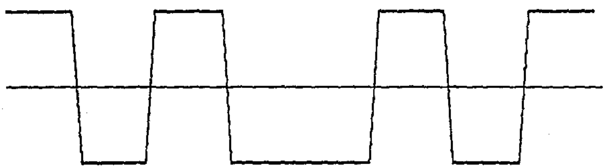
Fig. 4.1a Funciones de Walsh



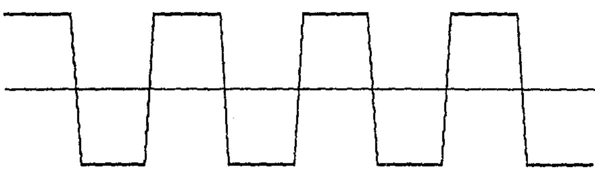
WAL(4,T)



WAL(5,t)

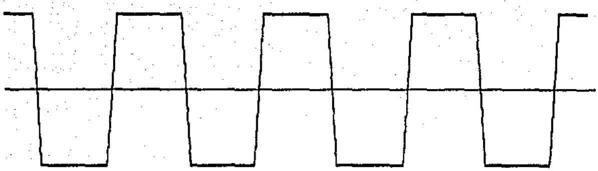


WAL(6,t)

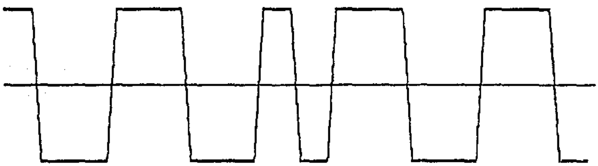


WAL(7,t)

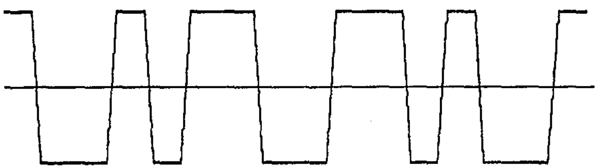
Fig. 4.1b Funciones de Walsh



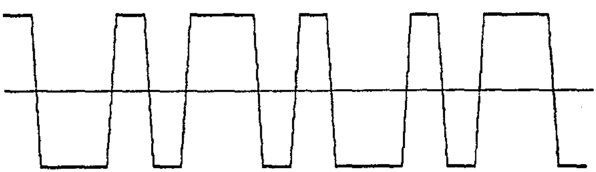
WAL(8,t)



WAL(9,t)

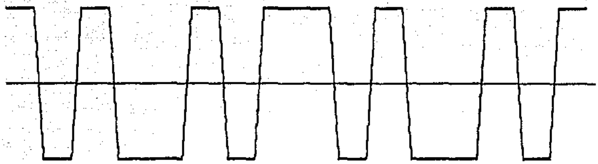


WAL(10,t)

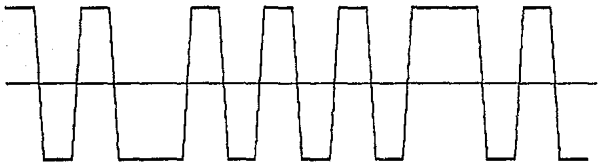


WAL(11,t)

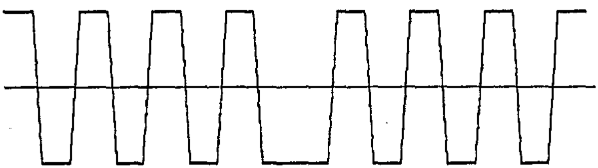
Fig. 4.1c Funciones de Walsh



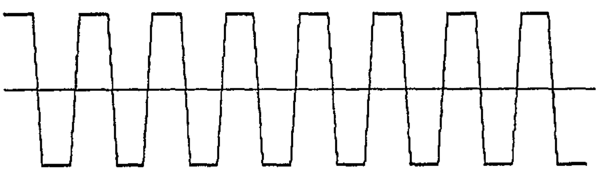
WAL(12,t)



WAL(13,t)

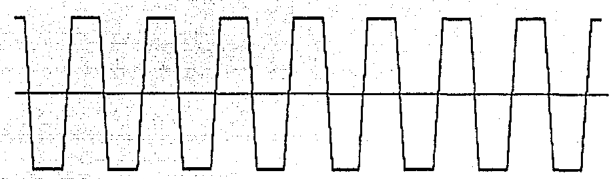


WAL(14,t)

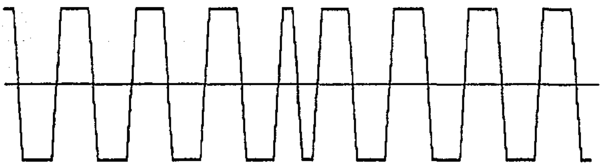


WAL(15,t)

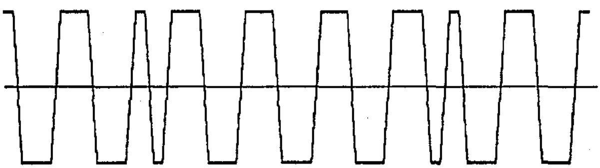
Fig. 4.1d Funciones de Walsh



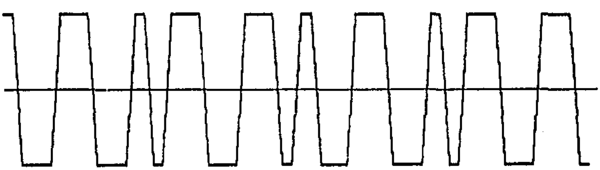
WAL(16,t)



WAL(17,t)

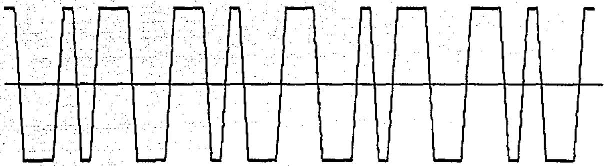


WAL(18,t)

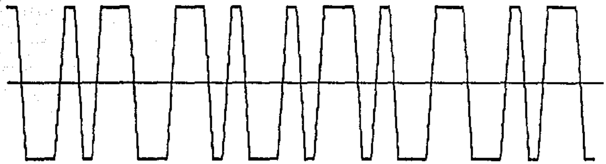


WAL(19,t)

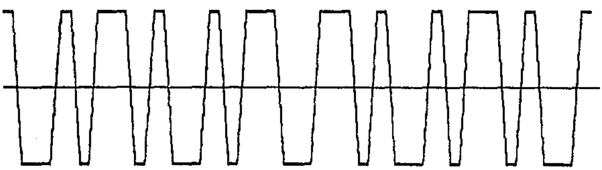
Fig. 4.1e Funciones de Walsh



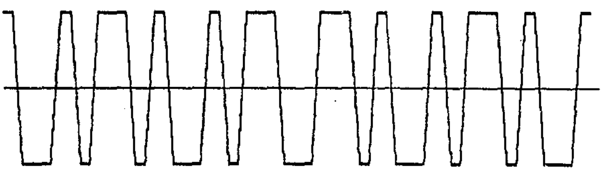
WAL(20,t)



WAL(21,t)

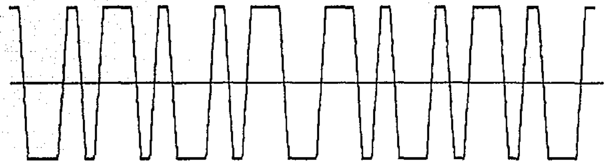


WAL(22,t)

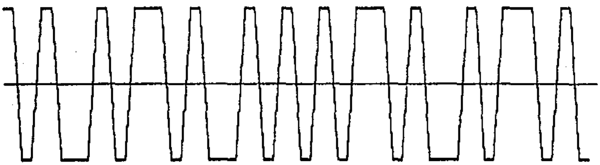


WAL(23,t)

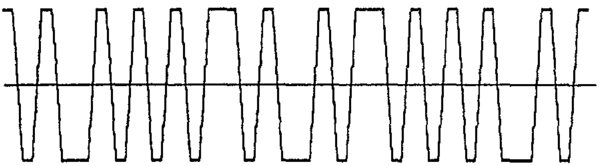
Fig. 4.1f Funciones de Walsh



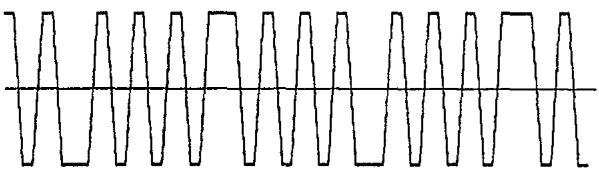
WAL(24,t)



WAL(25,t)



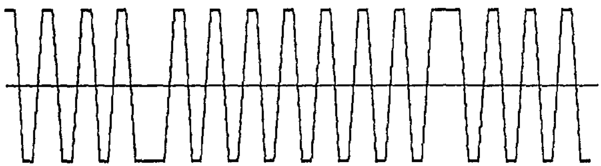
WAL(26,t)



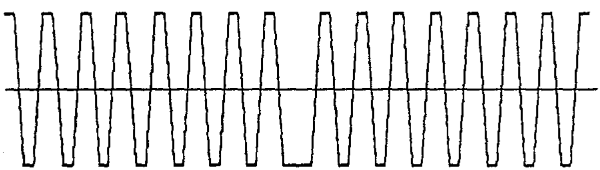
WAL(27,t)

Fig. 4.1g Funciones de Walsh

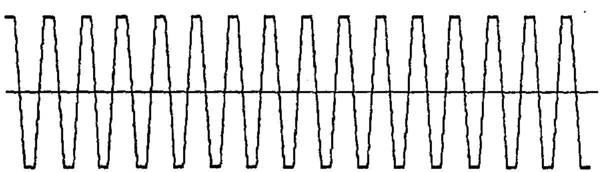
WAL(28,t)



WAL(29,t)



WAL(30,t)



WAL(31,t)

Fig.4.1h Funciones de Walsh

Otra alternativa de clasificación es la de Harmuth, que clasifica las funciones de Walsh en términos de Simetría non y par:

$$\begin{aligned} WAL(2k, t) &= CAL(k, t) \\ WAL(2k-1, t) &= SAL(k, t) \end{aligned} \qquad \text{Ecs. 4.2}$$

donde: $k = 0,1,2...$

Estas funciones tienen su contraparte en las funciones seno y coseno (Tabla 7.2).

El orden de las funciones descritas anteriormente se le denomina orden secuencial (Harmuth) u orden de Walsh-Kaczmarz, definido a partir del número de cruces por cero de la función por período de tiempo, el cual dependiendo si el número es par o non la función es CAL(k,t) o SAL(k,t) respectivamente, por lo tanto las funciones CAL tienen simetría par y las funciones SAL simetría non.

4.4 ORDENES Y RELACIONES ENTRE CAL (k,t) y SAL(k,t)

Aunque para las funciones seno y coseno, existe una relación muy sencilla entre ellas, como lo es el teorema de desfasamiento, para las funciones CAL(k,t) y SAL(k,t), la relación no es tan sencilla. Estas funciones están relacionados mediante la siguiente ecuación:

$$CAL(k, t+to) = SAL(k,t) \qquad \text{Ec. 4.3}$$

donde: $t_o = (-1)^{q+1} * 2^{-(r+2)}$

$k = (2^r)(2q + 1)$ $\qquad r,q = 0,1,2,.........$

A continuación se presenta una tabla para k en términos de r y q donde k = [1,2...8].

k	r	q
1	0	0
2	1	0
3	0	1
4	2	0
5	0	2
6	1	1
7	0	3
8	3	0

Tabla 4.1 Tabla de relación entre k, r y q.

Otro tipo de orden es el de orden natural (orden normal, orden binario, orden diádico, u orden de Paley). Este orden es el que se obtiene al generar las funciones de Walsh por medio de funciones de Rademacher, y se denota por la siguiente función:

$$PAL\,(\,k\,,\,t\,)$$

4.5 FORMAS DE OBTENER LAS FUNCIONES DE WALSH.

Las funciones de Walsh se pueden obtener mediante los siguientes métodos:

a) Por medio de Ecuaciones en diferencias.
b) Del producto de funciones de Rademacher[3].
c) De matrices de Hadamard.

a) Ecuaciones en Diferencias.

Considerando que la base de tiempo[4] es de $-\frac{1}{2} \leq t \leq \frac{1}{2}$, las funciones de Walsh se pueden obtener de la función de Walsh anterior por medio de la siguiente ecuación:

---

[3]Ver apéndice I
[4] Fase de Hadamard.

$$WAL(2j+q,t) = (-1)^{[j/2]+q}\ [WAL(j,2t) + (-1)^{j+q}\ WAL(j,2(t-½))] \quad \text{Ec.4.4}$$

donde: $q = 0,1$
$j = 0,1,2...$

Para N puntos ($N = 2^p$) discretos donde $n = 0,1,2 \ldots N\text{-}1$ se tiene que:

$$WAL(2j+q,n) = (-1)^{[j/2]+q}\ [WAL\ (j,2n) + (-1)^{j+q}\ WAL\ (j,2(n\text{-}N/2))]$$

Ec. 4.5

Para empezar se considera que $WAL(0,t) = 1$ en toda la base de tiempo (ver figura 4.1a).

b) Del producto de funciones de Rademacher.

Las funciones de Walsh en orden natural pueden ser obtenidas del producto de funciones de Rademacher, de la siguiente forma:

$$PAL(n,t) = \prod_{i=1}^{m} b_i\ R(i,t) \quad \text{Ec. 4.6}$$

donde n se expresa como un número binario, tal que:

$$n = b_m 2^m + b_{m-1} 2^{m-1} + \ldots + b_1 2^1 + b_0 2^0 \quad \text{Ec. 4.7}$$

donde: $b_i = 0,1$

Por ejemplo:

Para obtener la función PAL(11,t) se tiene que:

$$n = 11_{10} = 1011_2 = 1*2^3+0*2^2+1*2^1+1*2^0$$

$$PAL(11,t) = R(4,t)\ R(2,t)\ R(1,t)$$

Para obtener funciones de WALSH en orden secuencial se tiene que:

$$WAL(v,t) = \prod_{i=1}^{m} g_i R(i,t) \qquad \text{Ec. 4.8}$$

donde: $v = (g_m g_{m-1} \cdots\cdots g_o)$
$g_i = b_i \oplus b_{i+1}$ donde bi se obtiene de la ec. 4.7
$\oplus$ Suma en módulo 2 (XOR)

Ejemplo:

Para obtener la función WAL(9,t):

$n = 9_{10} = 1001_2$

$WAL\ (9,t) = R(4,t)\ R(3,t)\ R(1,t)$

c) De matrices de Hadamard

Las matrices de Hadamard son matrices cuadradas cuyos elementos sólo tienen valores de 1 y -1, y los renglones y columnas son ortogonales entre si.

La matriz de Hadamard elemental tiene orden 2 y se expresa como:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \text{Ec. 4.9}$$

Las siguientes matrices se pueden obtener con la siguiente fórmula recursiva:

$$H_N = H_{N/2} \otimes H_2 \qquad \text{Ec. 4.10}$$

donde: $\otimes$ representa el producto Kronecker o producto directo.
N debe ser potencia de dos ($N = 2^p$, $p = 2,3,4...$)

El producto de Kronecker se obtiene remplazando cada elemento de la matriz $H_{N/2}$ por la matriz H_2. Por ejemplo:

$$H_4 = H_2 \otimes H_2$$

$$H_4 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix}$$

Y para $H_8 = H_4 \otimes H_2$

$$= \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{vmatrix} = \begin{matrix} PAL(0,t) \\ PAL(4,t) \\ PAL(2,t) \\ PAL(6,t) \\ PAL(1,t) \\ PAL(5,t) \\ PAL(3,t) \\ PAL(7,t) \end{matrix}$$

La relación que existe entre las matrices de Hadamard y las funciones de Walsh se puede observar en la matriz H_8. Estas funciones tienen fase positiva[5] y un orden conocido como de Kronecker o orden Lexicográfico (Beauchamp,1975).

Otras formas de obtener las funciones de Walsh es por síntesis booleana (Beauchamp, 1975) o mediante la siguiente ecuación:

$$WAL(n,t) = \mathrm{sign}\left[(\mathrm{sen}(2\pi t))^{b_0} \cdot \prod_{k=1}^{m} \left(\cos\left(2^k \pi t\right)^{b_k} \right) \right] \qquad \text{Ec. 4.11}$$

Esta función se utilizó en el siguiente programa para obtener funciones de Walsh para N puntos.

[5] que está en el rango de [0,1], cuando está de [-½, ½] se nombra fase de Hadamard

FUNCION:

```
'****************************************
'SUBRUTINA PARA CALCULAR FUNCIONES DE WALSH
'****************************************

PRINT "NUMERO DE FUNCION DE WALSH"; : INPUT W
PRINT "NUMERO DE PUNTOS"; : INPUT N
E = W
NN = LOG(W) / LOG(2)
FOR I = 0 TO NN
   B(I) = E - INT(E / 2) * 2
   E = INT(E / 2)
NEXT I
FOR J = 0 TO N - 1
MM = 1
    FOR I = 1 TO NN
     MM = MM * COS(2 ^ I * PI * J / (N - 1)) ^ B(I)
    NEXT I
H(J + 1) = SGN((SIN(2 * PI * J / (N - 1)) ^ B(0) * MM))
NEXT J
H(1) = H(2)
H(J) = H(J - 1)
RETURN
```

4.6 DEFINICION. TRANSFORMADA DE WALSH

La Transformada de Walsh se define como (Beauchamp,1975):

$$W(k) = \int_0^1 f(t)\ WAL(k,t)\ dt \qquad 0 \leq t \leq 1 \qquad \text{Ec.4.11}$$

donde: f(t) es la función de tiempo
WAL(k,t) es la función k-ésima de WALSH

Ejemplo 4.1:

La Transformada de Walsh de la función pulso definida como:

$$\Pi(t) = \begin{cases} 0, & t < \frac{1}{4} \\ 1, & \frac{1}{4} < t < \frac{3}{4} \\ 0, & t > \frac{3}{4} \end{cases}$$

se expresa como:

$$W(k) = \int_0^1 f(t)\ WAL(k,t)\ dt$$

$$W(k) = \int_0^1 \Pi(t)\ WAL(k,t)\ dt$$

donde:

$$W(0) = \int_{1/4} WAL(0,t)\ dt = 0.5$$

$$W(1) = \int_{1/4}^{3/4} WAL(1,t)\ dt = 0$$

$$W(2) = \int_{1/4}^{3/4} WAL(2,t)\ dt = -0.5$$

$$W(k) = \int_{1/4}^{3/4} WAL(k,t)\ dt = 0 \qquad \text{para } k > 2$$

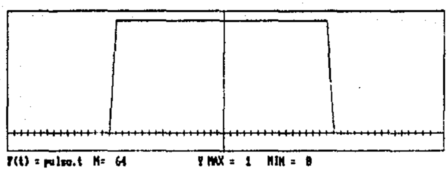


Fig. 4.2 Función pulso

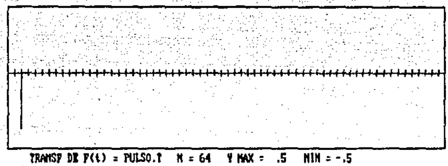


Fig. 4.3 Transformada de Walsh de la función Pulso.

Ejemplo 4.2

La Transformada de Walsh de la función $\cos\omega_0 t$

$$W(k) = \int_{½}^{½} \cos(\omega_0 t)\, WAL(k,t)\, dt$$

$$W(0) = \int_{½}^{½} \cos(\omega_0 t)\, WAL(0,t)\, dt = 0$$

$$W(1) = \int_{½}^{½} \cos(\omega_0 t)\, WAL(1,t)\, dt = 0.03$$

$$W(2) = \int_{½}^{½} \cos(\omega_0 t)\, WAL(2,t)\, dt = 0.6361$$

$$W(3) = \int_{½}^{½} \cos(\omega_0 t)\, WAL(3,t)\, dt = 0$$

$$W(4) = \int_{½}^{½} \cos(\omega_0 t)\, WAL(4,t)\, dt = 0$$

$$\vdots$$

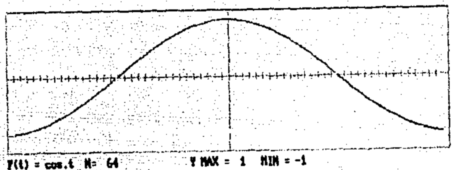


Fig. 4.4 Función $\cos\omega_0 t$

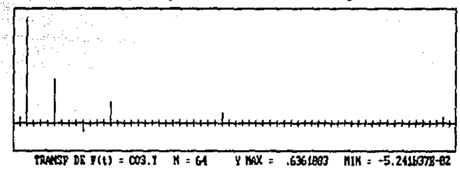


Fig. 4.5 Transformada de la función $\cos\omega_0 t$

4.7 PROPIEDADES DE LA TRANSFORMADA DE WALSH.

PROPIEDAD	WALSH
Ortogonalidad	$\sum_{t=0}^{N-1} WAL(m,t)\, WAL(n,t)\, dt = \begin{cases} N & \text{si } n = m \\ 0 & \text{si } n \neq m \end{cases}$
Simetría	$WAL(n,i) = WAL(i,n)$
Teorema de la multiplicación	$WAL(k,i)WAL(p,i) = WAL[(k\oplus p),i]$ $CAL(k,i)CAL(p,i) = CAL[(k\oplus p),i]$ $SAL(k,i)CAL(p,i) = SAL[(p\oplus(k-1))+1,i]$ $CAL(k,i)SAL(p,i) = SAL[(k\oplus(p-1))+1,i]$ $SAL(k,i)SAL(p,i) = CAL[(k-1)\oplus(p-1),i]$
Convolución Dyadica	$x_i * y_i = 1/N \sum_{i=0}^{N-1} x_i y_{(\tau\oplus i)}$
Autocorrelación Dyadica	$Rw(\tau) = 1/N \sum_{i=0}^{N-1} x_i x_{(\tau\oplus i)}$

Tabla 4.2

4.8 TRANSFORMADA INVERSA

La Transformada Inversa de Wash se puede escribir como:

$$f(t) = \sum_{k=0}^{\infty} W(k)\ WAL(k,t) \qquad \text{Ec. 4.13}$$

donde: $f(t) \Leftrightarrow W(k)$ es un par de transformadas de Walsh.

4.9 SERIES DE WALSH

De manera similar a la serie de Fourier (Ec. 2.10) una función f(t) se representa en términos de funciones de Walsh como:

$$f(t) = a_0 WAL(0,t) + \sum_{n=1}^{N-1} a_n WAL(n,t) \qquad \text{Ec. 4.14}$$

donde:

$$\frac{a_0}{2} = \frac{1}{T}\int_0^T f(t)\ WAL(0,t)\ dt \qquad \text{Ec. 4.15}$$

$$a_n = \frac{1}{T}\int_0^T f(t)\ WAL(n,t)\ dt \qquad \text{Ec. 4.16}$$

$$WAL(0,t) = 1 \qquad \text{para } [0,1]$$

Las ecuación 4.16 en términos de funciones CAL y SAL (ecs. 4.2) se representan como:

$$b_j = \frac{1}{T}\int_0^T f(t)\ CAL(j,t)\ dt \qquad \text{Ec. 4.17}$$

$$a_i = \frac{1}{T}\int_0^T f(t)\ SAL(i,t)\ dt \qquad \text{Ec. 4.18}$$

Sustituyendo Ecs 4.17 y 4.18 en la Ec. 4.14, la función f(t) se puede representar como:

$$f(t) = a_0 WAL(0,t) + \sum_{i=1}^{\frac{N}{2}} \sum_{j=1}^{\frac{N}{2}-1} \left[a_i SAL(i,t) + b_j CAL(j,t) \right] \qquad \text{Ec. 4.19}$$

4.10 TRANSFORMADA DISCRETA DE WALSH.

Para utilizar la transformada con datos discretos o numéricos es conveniente utilizar la Transformada Discreta de Walsh, que se puede definir como:

$$W(k) = {}^{1}/_{N} \sum_{i=0}^{N-1} f(nT)\ WAL\ (k,i) \qquad \text{Ec. 4.20}$$

donde: $n = 0,1,2.... N-1$

La Transformada Inversa Discreta se define como:

$$f(nT) = \sum_{k=0}^{N-1} W(k)\ WAL\ (k,nT) \qquad \text{Ec. 4.21}$$

4.11 RELACION ENTRE LAS TRANSFORMADAS DE WALSH Y FOURIER.

De las ecuaciones 4.20 y 2.15 se puede obtener la Transformada de Walsh a partir de la transformada de Fourier como:

$$W(k) = {}^{1}/_{N} \sum_{n=0}^{N-1} F(k) [\sum_{i=0}^{N-1} WAL(k,i)\ e^{2\pi jnk/N}] \qquad \text{Ec. 4.22}$$

A su vez la transformada de Fourier puede ser obtenida de la Transformada de Walsh como:

$$F(k) = {}^{1}/_{N} \sum_{k=0}^{N-1} W(k)[\sum_{n=0}^{N-1} WAL(k,nT)\ e^{-j2\pi nk/N}] \qquad \text{Ec. } 4.23$$

4.12 TRANSFORMADA RAPIDA DE WALSH (FWT[6]).

El algoritmo utilizado fue el de Beauchamp y reduce al mismo número de operaciones que las Transformadas de Fourier y Hartley ($N\log_2 N$), los datos de entrada se consideran igual que los programas anteriores:

```
WALSH:
    '*********************************************
    'SUBRUTINA PARA CALCULAR LA TRANSFORMADA RAPIDA DE WALSH
    '*********************************************
  N2 = N / 2: P = LOG(N) / LOG(2)
  FOR L = 1 TO P
   NY = 0: NZ = 2 ^ (L - 1): NZI = 2 * NZ: NZN = N / NZI
   FOR I = 1 TO NZN
    NX = NY + 1: NY = NY + NZ:JS = (I - 1) * NZI: JD = JS + NZI + 1
    FOR J = NX TO NY
      JS = JS + 1:J2 = J + N2
      X(JS) = F(J) + F(J2):JD = JD - 1
      X(JD) = F(J) - F(J2)
    NEXT J
   NEXT I
   FOR K = 1 TO N
     F(K) = X(K)
   NEXT K
  NEXT L
  NEXT K
RETURN
```

[6]Fast Walsh Transform

CAPITULO V. TRANSFORMADA DE HAAR

5.1 INTRODUCCION Y ANTECEDENTES.

La Transformada de Haar junto con Walsh, forman parte del grupo de Transformadas que descomponen una señal en un conjunto de funciones rectangulares. Históricamanete las funciones de Haar fueron descritas por el matemático Húngaro Alfred Haar en 1910. Estas funciones también forman un conjunto completo de funciones rectangulares pero con amplitudes diferentes dependiendo del número de función de que se trate.

5.2 FUNCIONES DE HAAR

Las funciones de Haar forman un conjunto ortogonal de funciones rectangulares con un sólo período. La amplitud A de estas funciones no tienen valor de amplitud único como el caso de las funciones de Walsh[1] , su valor está en función de:

$$A = (\sqrt{2})^{p} \qquad \text{Ec. 5.1}$$

donde: $p = 1,2\ldots\ldots\ldots$

Las funciones de Haar se pueden representar en el intervalo $0 \leq t \leq 1$ como:

$$HAR(n, t)$$

donde: n identifica el número de la función
t es la base de tiempo.

[1] Ver capitulo IV

Por tratarse de funciones ortogonales[2], deben de cumplir que:

$$\int_0^1 \text{HAR}(m,t) * \text{HAR}(n,t) = \begin{cases} 1 & \text{si } n = m \\ 0 & \text{si } n \neq m \end{cases} \qquad \text{Ec. 5.2}$$

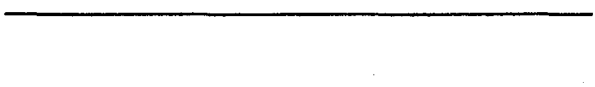
5.3 FORMA DE OBTENER LAS FUNCIONES DE HAAR

Las funciones de Haar se pueden obtener por medio de la siguiente ecuación.

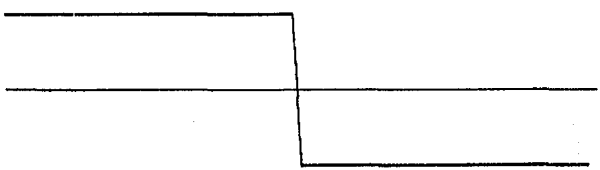
$$\text{HAR}(n,t) = \begin{cases} \sqrt{2^p} & \text{para } n/2^p \leq t \leq (n+\frac{1}{2})/2^p \\ -\sqrt{2^p} & \text{para } (n+\frac{1}{2})/2^p \leq t \leq (n+1)2^p \\ 0 & \text{en cualquier otro caso.} \end{cases} \qquad \text{Ec. 5.3}$$

donde: $p = 1,2\ldots.. \; n = 0,1\ldots\ldots 2^p - 1$

En las figuras 5.1x se muestran las primeras 32 funciones de Haar[3] y la amplitud A de cada una:



HAR(0,t), A=1

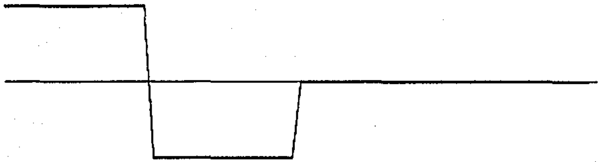


HAR(1,t), A= +-1

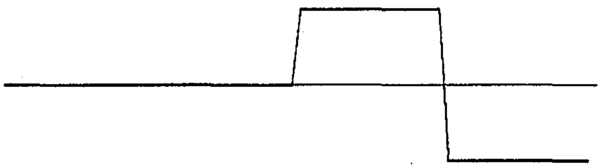
Fig. 5.1a Funciones de Haar

[2] Ver capítulo I

[3] La función HAR(0,t) tiene el valor de 1 para $0 \leq t \leq 1$



HAR(2,t),A = +-√2

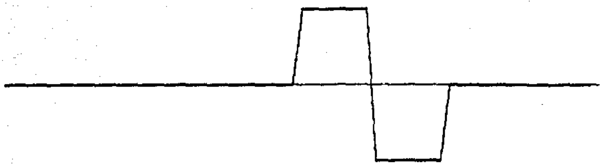


HAR(3,t), A = +-√2

HAR(4,t), A = +-2

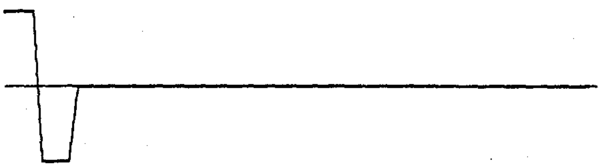
HAR(5,t),A = +-2

Fig. 5.1b Funciones de Haar

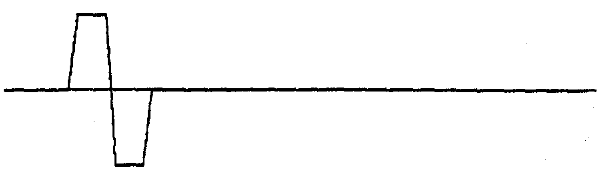


HAR(6,t),A= +-2

HAR(7,t),A= +-2

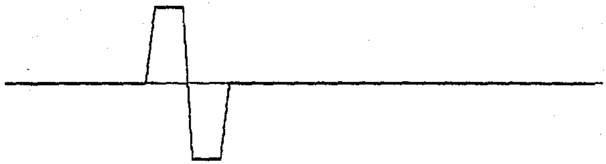


HAR(8,t),A= +-2√2

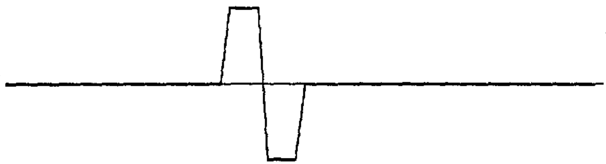


HAR(9,t),A= +-2√2

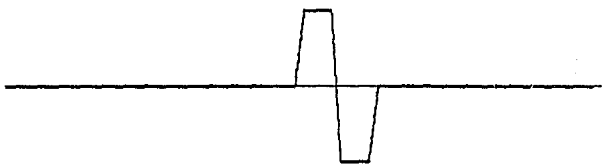
Fig. 5.1c Funciones de Haar



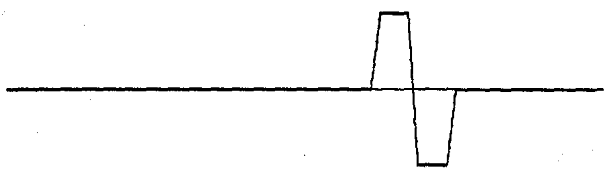
HAR(10,t), A = +-2√2



HAR(11,t), A = +-2√2

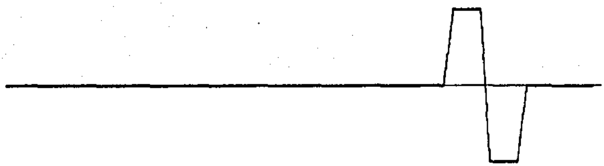


HAR(12,t),A = +-2√2



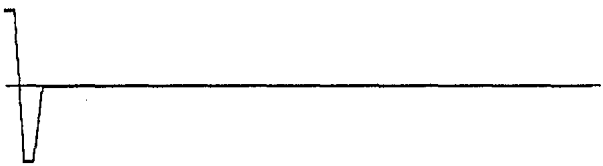
HAR(13,T),A = +-2√2

Fig. 5.1d Funciones de Haar



HAR(14,t),A = +-2√2

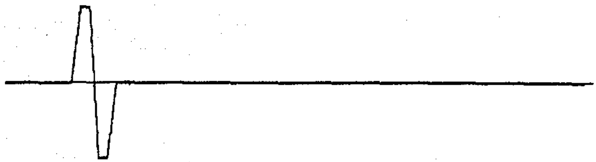
HAR(15,t),A = +-2√2



HAR(16,T),A = +-4

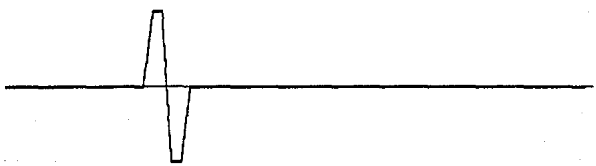
HAR(17,t),A = +-4

Fig. 5.1e Funciones de Haar

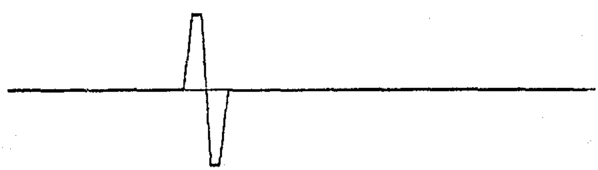


HAR(18,t), A = +-4

HAR(19,t), A = +-4

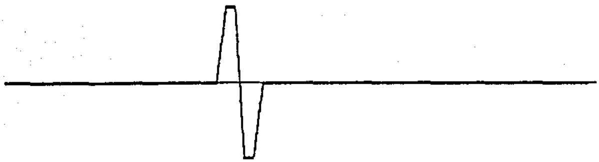


HAR(20,t), A = +-4

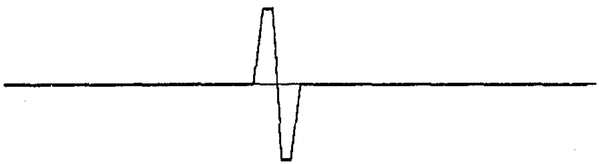


HAR(21,t), A = +-4

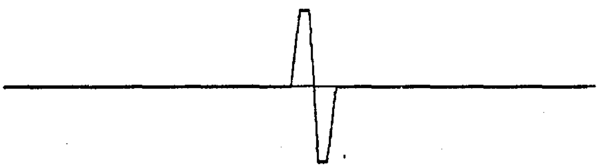
Fig. 5.1f Funciones de Haar



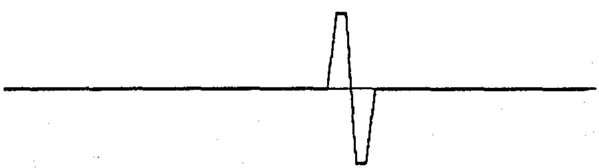
HAR(22,t), A= +-4



HAR(23,T), A= +-4



HAR(24,t), A= +-4

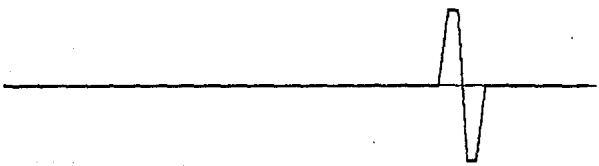


HAR(25,t), A= +-4

Fig. 5.1 g Funciones de Haar

HAR(26,t), A= +-4

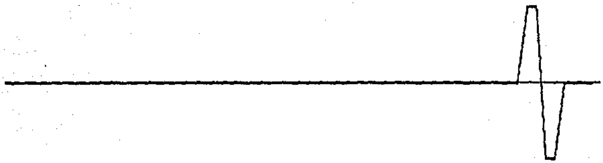
HAR(27.t), A= +-4



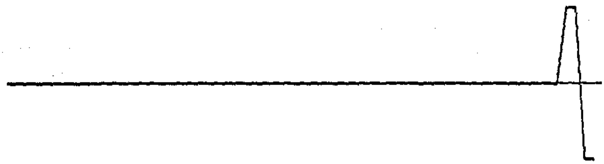
HAR(28,t), A= +-4

HAR(29,t), A= +-4

Fig. 5.1h Funciones de Haar



HAR(30,t), A= +-4



HAR(31,t), A= +-4

Fig. 5.1 i Funciones de Haar

De las figuras 5.1x se puede observar que las funciones de HAAR forman subconjuntos donde se tiene en común la misma amplitud y período pero con un desfasamiento de T unidades, donde T es el período de la función rectangular (ver tabla 5.1) (Shore,1973).

Otra característica que tienen las funciones de Haar, es la de convergencia (que es superior a la de Walsh), que permite representar discontinuidades en un segmento de la función con un número "pequeño" de términos.

Función	Grupo
HAR(0,t)	1
HAR(1,t)	2
HAR(2,t) ,HAR(3,t)	3
HAR(4,t),HAR(5,t) HAR(6,t),HAR(7,t)	4
HAR(8,t),HAR(9,t) HAR(10,t),HAR(11,t) HAR(12,t)HAR(13,t) HAR(14,t),HAR(15,t)	5

Tabla 5.1 Grupos de funciones de Haar con la misma amplitud

A continuación se presenta un programa para obtener funciones de Haar[4] para N puntos:

```
FUNCION:
        '*************************************************************
        '     SUBRUTINA PARA GENERAR FUNCIONES DE HAAR
        '*************************************************************
        PRINT : PRINT "NUMERO DE LA FUNCION DE HAAR A GENERAR "; : INPUT W
        PRINT "NUMERO DE PUNTOS "; : INPUT N
        PRINT "NOMBRE DEL ARCHIVO"; : INPUT NOMBRE$
        LW = INT(LOG(W) / LOG(2)):R = W - 2 ^ LW
        L1 = N * R / 2 ^ LW:L2 = N * (R + 1 / 2) / 2 ^ LW
        FOR L = L1 TO L2 - 1
         H(L + 1) = (2 ^ .5) ^ LW
        NEXT L
        L3 = N * (R + 1) / 2 ^ LW
        FOR L = L2 TO L3 - 1
         H(L + 1) = -(2 ^ .5) ^ LW
        NEXT L
        RETURN
```

Existe otro tipo de notación para las funciones de Haar (Shore, 1973):

$$F_n^m(t)$$

donde: n representa el conjunto de funciones con la misma amplitud
m representa el número de función dentro del conjunto.

[4] Se recomienda que N sea potencia de dos para que tenga relación con la Transformada Rápida de Haar (FHAT), donde los datos deben ser también potencia de 2.

por ejemplo:

$$HAR(5,T) = F_3^2(t) \qquad \text{Ec. 5.4}$$

5.4 DEFINICION.TRANSFORMADA DE HAAR.

La Transformada de Haar se define como:

$$HA(n) = \int_0^1 f(t)\ HAR(n,t)\ dt \qquad \text{Ec 5.5}$$

donde los límites de la integral corresponden a la base de tiempo $0 \leq t \leq 1$.

Ejemplo 5.1:

La Transformada de Haar para la función pulso definida como:

$$\Pi(t) = \begin{cases} 0, & t < 1/4 \\ 1, & 1/4 < t < 3/4 \\ 0, & t > 3/4 \end{cases}$$

se puede expresar como:

$$HA(n) = \int_0^1 \Pi(t)\ HAR(n,t)\ dt$$

$$HA(0) = \int_{1/4}^{3/4} HAR(0,t)\ dt = 0.5$$

$$HA(1) = \int_{1/4}^{3/4} HAR(1,t)\ dt = 0$$

$$HA(2) = \int_{1/4}^{3/4} HAR(2,t)\, dt = -0.3535534$$

$$HA(3) = \int_{1/4}^{3/4} HAR(3,t)\, dt = 0.3535534$$

$$HA(k) = \int_{1/4}^{3/4} HAR(k,t)\, dt = 0 \qquad \text{para } k > 3$$

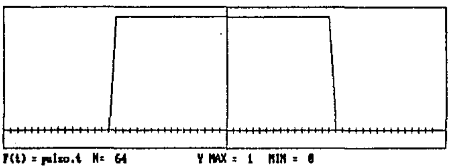


Fig. 5.2 Función Pulso

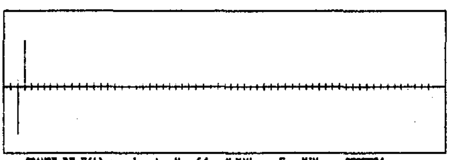


Fig. 5.3 Transformada de Haar de la Función Pulso.

Ejemplo 5.2:

La Transformada de Haar de la función $\cos\omega_0 t$ se puede obtener como:

$$HA(n) = \int_{\frac{1}{2}}^{\frac{1}{2}} \cos(\omega_0 t)\ HAR(n,t)\ dt$$

$$HA(0) = \int_{\frac{1}{2}}^{\frac{1}{2}} \cos(\omega_0 t)\ HAR(0,t)\ dt = 0$$

$$HA(1) = \int_{\frac{1}{2}}^{\frac{1}{2}} \cos(\omega_0 t)\ HAR(1,t)\ dt = 0.0312$$

$$HA(2) = \int_{\frac{1}{2}}^{\frac{1}{2}} \cos(\omega_0 t)\ HAR(2,t)\ dt = 0.4497966$$

$$HA(3) = \int_{\frac{1}{2}}^{\frac{1}{2}} \cos(\omega_0 t)\ HAR(3,t)\ dt = -0.4497966$$

$$HA(4) = \int_{\frac{1}{2}}^{\frac{1}{2}} \cos(\omega_0 t)\ HAR(4,t)\ dt = 0.1252703$$

$$\vdots$$

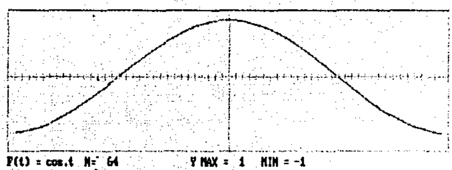


Fig. 5.4 Función $\cos\omega_0 t$

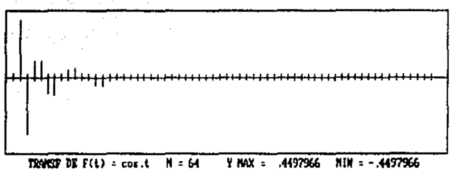


Fig. 5.5 Transformada de la función $\cos\omega_0 t$

5.5 TRANSFORMADA INVERSA DE HAAR.

La Transformada Inversa de Haar se define como:

$$f(t) = \sum_{n=0}^{\infty} C_n \, HAR(n,t) \qquad \text{Ec.5.6}$$

donde: C_n se puede obtener de la Ec. 5.5

5.6 RELACIONES ENTRE LAS FUNCIONES DE HAAR Y WALSH.

Las funciones de Haar se pueden obtener mediante las funciones de Walsh por medio de la siguiente relación (Beauchamp,1975):

$$HAR(2^p+n,t) = 1/(2\sqrt{2p}) \sum_{k=0}^{(2^{p+1}-1)} [WAL(2n,k/2^{p+1})-WAL(2n+1,k/2^{p+1})]WAL(k,t)$$

Ec. 5.7

5.7 SERIES DE HAAR.

Al igual que con las otras transformadas, una función f(t) se puede expresar en función de una serie infinita de funciones de HAAR:

$$f(t) = c_o + \sum_{n=1}^{\infty} c_n \, HAR(n,t) \qquad \text{Ec. 5.8}$$

donde: c_0 y c_n se pueden obtener de la ec. 5.5.

5.8 TRANSFORMADA DISCRETA DE HAAR

La Transformada Discreta de Haar y su inversa se definen como:

$$X_n = {}^1/_N \sum_{n=0}^{N-1} x_i \, HAR(n, {}^i/_N) \qquad \text{Ec.5.9}$$

$$x_i = \sum_{n=0}^{N-1} X_n \, HAR(n, {}^i/_N) \qquad \text{Ec.5.10}$$

donde: i , n = 0,1....N-1
N es el número de datos a analizar.

Las ecuaciones 5.9 y 5.10 se pueden escribir en forma matricial como:

$$\mathbf{X} = {}^1/_N \, H \, \mathbf{x} \qquad \text{Ec.5.11}$$

$$\mathbf{x} = H^{-1} \mathbf{X} \qquad \text{Ec. 5.12}$$

donde: H y H^{-1} son la matriz directa e inversa de funciones de HAAR

5.9 TRANSFORMADA RAPIDA DE HAAR (FHAT)[5]

Al igual que las transformadas anteriores el vector de salida F(f) se utiliza para guardar datos de entrada antes de obtener la Transformada. La Transformada Rápida de Haar realiza 2(N-1) operaciones en el momento de hacer los cálculos.

HAAR:

```
      ****************************************************************
      ' SUBRUTINA PARA CALCULAR LA TRANSFORMADA DE HAAR
      ****************************************************************

  P = LOG(N) / LOG(2)
  FOR I = 1 TO P
   L = P + 1 - I
   L2 = 2 ^ (L - 1)
   FOR Z = 1 TO 2 * L2
      I(Z) = F(Z)
   NEXT Z
   FOR J = 1 TO L2
      L3 = L2 + J
      JJ = 2 * J - 1
      F(J) = I(JJ) + I(JJ + 1)
      F(L3) = I(JJ) - I(JJ + 1)
   NEXT J
  NEXT I
  FOR I = 1 TO N
      IF I > 1 THEN PP = INT(LOG(I - 1) / LOG(2))
      F(I) = F(I) / N * ((2 ^ .5) ^ PP)
  NEXT I
RETURN
```

[5] Ver apéndice 5 Fast Haar Transformation , FHAT

CAPITULO VI - METODO DE MAXIMA ENTROPIA

6.1 INTRODUCCION

La Transformada de Fourier (FFT) "tradicionalmente" es un algoritmo que se ha utilizado en una gran cantidad de áreas y del cual se puede contar con bastante literatura en cuanto a algoritmos programas, etc. No existe algún programa o paquete comercial de Procesamiento Digital de Señales (DSP) que no contenga la FFT como subrutina, función ó instrucción, pero como se describió en el Capítulo II, la FFT tiene ciertas limitaciones, como son: errores de pseudointerferencia, sensibilidad al ruído, etc.

Para señales en las cuales FFT tiene ciertas limitaciones existen otros algoritmos (Kay & Marple, 1981) alternos a Fourier. En este capítulo se analiza el método de Máxima Entropía como herramienta en el análisis de espectro.

6.2 ANTECEDENTES

El método de Máxima Entropía (MEM) originalmente fue desarrollado para el procesamiento de datos Geofísicos (Kay & Marple, 1981). Este método fue presentado por primera vez en el trabajo "Maximum Entropy Spectral Analysis" y fue desarrollado por John Burg en 1967. El resumen del trabajo original es el siguiente (Robinson, 1982):

" El método digital convencional para estimar el espectro de Potencia de una función cuya autocovarianza se conoce, asume que la función de autocorrelación es cero para todos los segmentos para los cuales no se puede estimar, y utiliza de cierta manera los segmentos conocidos para reducir los efectos de truncamiento de la función de autocovarianza. El método que se presenta en este trabajo en vez de utilizar todos los segmentos conocidos de la función utiliza los segmentos (diferentes de cero) para estimar los no conocidos. El principio de estimación que se utilizó en particular, es que el espectro estimado debe ser el más aleatorio o el de mayor entropía para cualquier espectro de potencia el cual es consistente con los datos medidos. Esta nueva técnica de análisis da mucho mayor resolución al espectro estimado, que el que se obtiene por técnicas convencionales con un pequeño incremento en el tiempo de cómputo. Las comparaciones ilustrarán su importancia".

El método de Máxima Entropía está basado en escoger que espectro corresponde al más aleatorio o a la más impredecible serie de tiempo cuya autocovarianza coincida con el conjunto de valores dado. Este método se conoce también con el nombre de Modelo de todos polos o Modelo Autoregresivo (AR) (Press,1986).

6.3 DEFINICION. METODO DE MAXIMA ENTROPIA (MEM)

El espectro de Potencia[1] se puede expresar de la siguiente forma (Press,1986):

$$P(f) = \frac{a_0}{\left|1 + \sum_{k=1}^{m} a_k * z^k\right|} \qquad \text{Ec. 6.1}$$

donde: m = es el número de polos
$z = e^{i2\pi f}$ $i = 0,1,2...$
$a_0 ... a_k$ son valores constantes.

Tomando en cuenta el teorema de Wiener-Khinchim, que dice:

"La Transformada de Fourier de la autocorrelación de una función es igual al espectro de potencia".

---

[1] Es la magnitud del espectro de Fourier elevado al cuadrado

El espectro de potencia se puede expresar como (Press,1986):

$$P(f)=\frac{a_0}{|1+\sum_{k=1}^{m} a_k *z^k|} = \sum_{j=-m}^{m} \Phi_j *z^j \qquad \text{Ec. 6.2}$$

donde: Φ_j es la autocorrelación de la función muestreada.

La Autocorrelación Φj se puede obtener por:

$$\Phi_j = \Phi_{-j} = \frac{1}{N+1-j}\sum_{i=0}^{N-j} c_i c_{i+j} \qquad \text{Ec. 6.3}$$

donde: $j = 0,1,2........N$
c_i representa los datos de la función

Para obtener los coeficientes $a_0, a_1,....a_k$ se tiene que cumplen con la siguiente relación (Matriz simétrica de Toeplitz):

$$\begin{vmatrix} \Phi_0 & \Phi_1 & \Phi_2 & \cdots & \Phi_M \\ \Phi_1 & \Phi_2 & \Phi_3 & \cdots & \Phi_{M-1} \\ \Phi_2 & \Phi_1 & \Phi_0 & \cdots & \Phi_{M-2} \\ \cdots & \cdots & & & \cdots \\ \Phi_M & \Phi_{M-1} & \Phi_{M-2} & \cdots & \Phi_0 \end{vmatrix} * \begin{vmatrix} 1 \\ a_1 \\ a_2 \\ \cdots \\ a_M \end{vmatrix} = \begin{vmatrix} a_0 \\ 0 \\ 0 \\ \cdots \\ 0 \end{vmatrix}$$

Ec. 6.4

de donde por el método de Burg y Andersen (Press, 1986) pueden ser obtenidos los valores de los coeficientes.

6.4 PROGRAMA EN COMPUTADORA

En el siguiente programa se presenta un método para obtener los coeficientes a_0 hasta a_k por el método de Burg y Andersen (Press,1986), como datos se da el número de polos y valores de entrada en el vector H(f). Este algoritmo no es una transformada rápida por lo que la cantidad de puntos a analizar puede ser cualquiera.

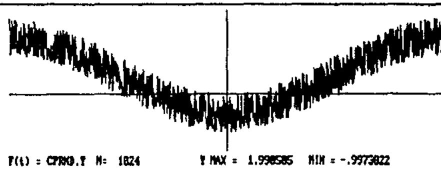
```
MEM :
  INPUT "Numero de polos"; MP
  INPUT "Incremento en frecuencia"; INC
  INPUT "Frecuencia final"; FF
  IF INC = 0 THEN INC = 1 / N
  IF FF = 0 THEN FF = 1
   P = 0
  FOR J = 1 TO N
   P = P + H(J) ^ 2
  NEXT J
  PM = P / N
  WK1(1) = H(1)
  WK2(N - 1) = H(N)
  FOR J = 2 TO N - 1
   WK1(J) = H(J)
   WK2(J - 1) = H(J)
  NEXT J
FOR K = 1 TO MP
   NUM = 0
   DENOM = 0
   FOR J = 1 TO N - K
    NUM = NUM + WK1(J) * WK2(J)
    DENOM = DENOM + WK1(J) ^ 2 + (WK2(J)) ^ 2
   NEXT J
   COF(K) = 2 * NUM / DENOM
   PM = PM * (1 - (COF(K)) ^ 2)
   FOR I = 1 TO K - 1
    COF(I) = WKM(I) - COF(K) * WKM(K - I)
   NEXT I
   IF K = M THEN GOTO MEM2
   FOR I = 1 TO K
    WKM(I) = COF(I)
   NEXT I
   FOR J = 1 TO N - K - 1
    WK1(J) = WK1(J) - WKM(K) * WK2(J)
    WK2(J) = WK2(J + 1) - WKM(K) * WK1(J + 1)
   NEXT J
NEXT K
MEM2:
        NP = FF / INC
        FOR IT = 1 TO NP
                fdt = INC * FF * IT
```

```
                theta = 2 * PI * fdt
                wpr = COS(theta)
                wpi = SIN(theta)
                wr = 1
                wi = 0
                sumr = 1
                sumi = 0
                FOR I = 1 TO MP
                        wtemp = wr
                        wr = wr * wpr - wi * wpi
                        wi = wi * wpr + wtemp * wpi
                        wrs = wr
                        wis = wi
                        sumr = sumr - COF(I) * wrs
                        sumi = sumi - COF(I) * wis
                        EVLMEM = PM / ((sumr) ^ 2 + (sumi) ^ 2)
                NEXT I
                F(IT) = EVLMEM
        NEXT IT
RETURN
```

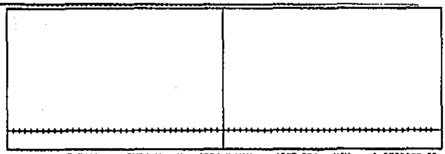
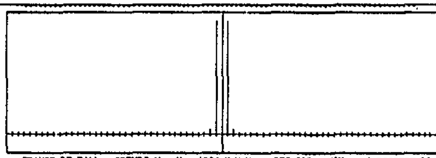
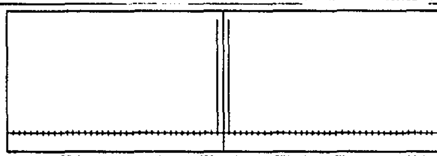
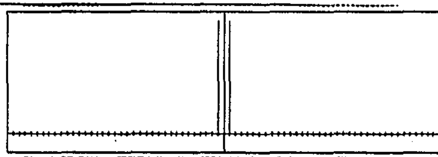
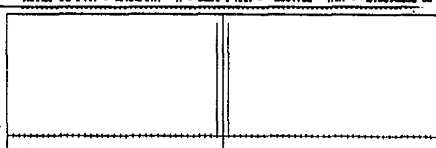
Los resultados obtenidos con este método muestran que con un valor de polos específico (diferente para cada función) es especialmente útil para funciones que tienen cierto valor de ruido, por ejemplo, los datos obtenidos en un experimento o alguna señal como puede ser la de voz. A diferencia de las otras transformadas, la cantidad de datos no tiene que ser potencia de dos, pero el tiempo de cálculo es mayor y se incrementa con la cantidad de datos y polos seleccionados.

6.5 EJEMPLOS

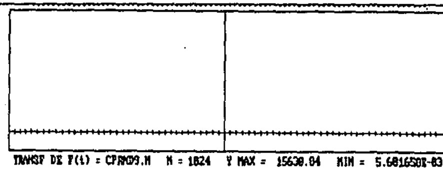
En la siguiente tabla se muestran varias funciones y su espectro con diferentes número de polos.

Función f(t)[2] cos(wt) + RND	

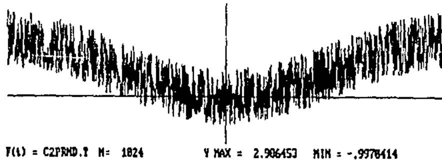
[2] La función "RND" representa el random de la computadora

Espectro con 50 polos	 TRANSF DE F(t) = CPRMD1.M N = 1024 Y MAX = 1005.894 MIN = 6.053937E-02
Espectro con 100 polos	 TRANSF DE F(t) = CPRMD2.M N = 1024 Y MAX = 250.805 MIN = 4.322973E-02
150 polos	 TRANSF DE F(t) = CPRMD3.M N = 1024 Y MAX = 252.225 MIN = 2.613524E-02
con 200 polos	 TRANSF DE F(t) = CPRMD4.M N = 1024 Y MAX = 286.702 MIN = 2.339023E-02
con 300 polos	 TRANSF DE F(t) = CPRMD6.M N = 1024 Y MAX = 810.3905 MIN = 8.948152E-03

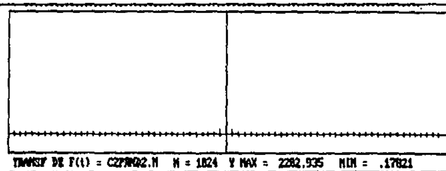
con 550 polos



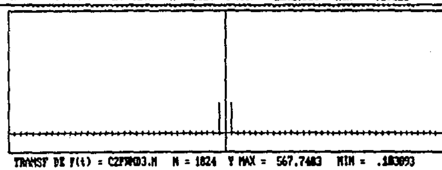
cos(wt) + 2*RND



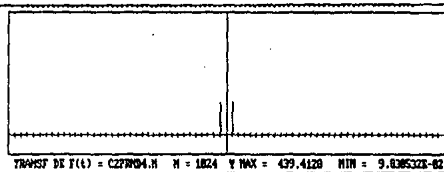
Espectro con 100 polos

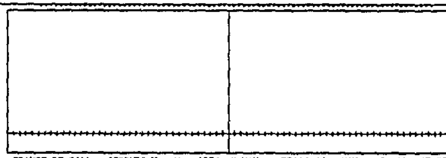


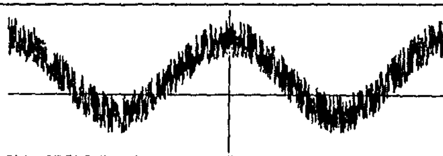
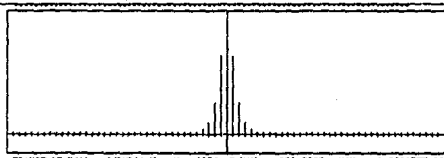
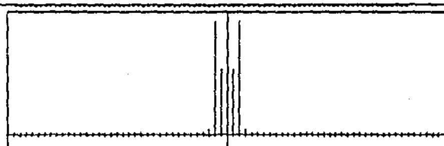
con 150 polos

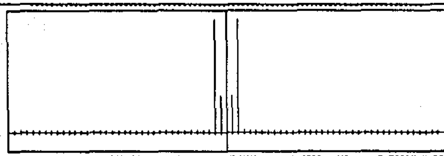
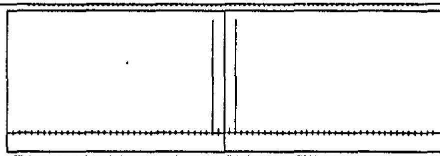
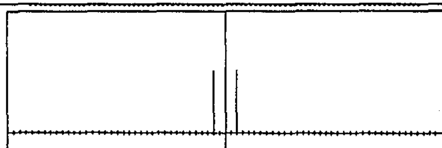


con 200 polos



con 300 polos	 TRANSF DE F(t) = C2PRND6.M N = 1024 Y MAX = 923.9366 MIN = 3.673074E-02
con 550 polos	 TRANSF DE F(t) = C2PRND9.M N = 1024 Y MAX = 50114.44 MIN = 2.206949E-02

cos(2wt) + RND	 F(t) = C4RND1.T N= 1024 Y MAX = 1.998279 MIN = -.982962
Espectro con 50 polos	 TRANSF DE F(t) = C4RND11.M N = 1024 Y MAX = 226.9837 MIN = 6.830677E-02
con 100 polos	 TRANSF DE F(t) = C4RND12.M N = 1024 Y MAX = 129.6409 MIN = 5.808613E-02

con 150 polos	 TRANSF DE F(t) = C4RMD13.M N = 1024 Y MAX = 299.4039 MIN = 2.598274E-02
con 200 polos	 TRANSF DE F(t) = C4RMD14.M N = 1024 Y MAX = 834.7363 MIN = 2.112468E-02
con 250 polos	 TRANSF DE F(t) = C4RMD15.M N = 1024 Y MAX = 1431.52 MIN = 1.610122E-02

CAPITULO VII.- COMPARACION ENTRE TRANSFORMADAS

En los capítulos II al VI se ha hecho una revisión de diferentes Transformadas tanto teórica, como numérica (programas en computadora), y en algunos casos se puede observar algunas relaciones y ventajas comunes que tienen entre si. En este capítulo se hace una comparación con respecto a características que se consideraron importantes para el análisis de espectro. Se hace hincapié en que algunas características de comparación no existen o no son análogas para todas las Transformadas.

Las comparaciones se harán a manera de tablas y gráficas con descripción y referencia a los capítulos, artículos, ecuaciones o autores correspondientes, por si se quiere mayor información sobre cada característica o fórmula desarrollada.

Varias de las fórmulas, programas o características fueron modificadas en cuanto a notación con respecto a las originales para tener un estándar en el momento de la comparación.

TRANSFORMADA	DEFINICION	TIPO DE FUNCIONES EN LAS QUE SE DESCOMPONE	ORTOGONALIDAD DE SUS FUNCIONES	NATURALEZA DE LOS NUMEROS QUE MANEJA
FOURIER	$F(f) = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi ft}\, dt$ Ec. 2.1	SENOIDAL $\cos wt - j \operatorname{sen} wt$	$\left.\begin{array}{l} = \int 2 \operatorname{sen}(2\pi nt) \operatorname{sen}(2\pi mt)\, dt \\ = \int 2 \operatorname{sen}(2\pi nt) \cos(2\pi mt)\, dt \\ = \int 2 \cos(2\pi nt) \operatorname{sen}(2\pi mt)\, dt \end{array}\right\} = 0 \text{ si } m \neq n$	Complejos
HARTLEY	$H(f) = \int_{-\infty}^{\infty} f(t) \operatorname{cas}(2\pi ft)\, dt$ Ec. 3.1	SENOIDAL $\operatorname{cas} wt = \cos wt + \operatorname{sen} wt$	Igual que Fourier	Reales
MAXIMA ENTROPIA	$P(F) = \sum_{j=-m}^{m} \Phi_j \cdot z^j$ Ec. 6.1	SENOIDAL	No comparable	Complejos
WALSH	$W(k) = \int_0^1 f(t)\, WAL(k,t)\, dt$ Ec. 4.11	RECTANGULAR WAL(n,t)	$\sum WAL(m,t)\, WAL(n,t)\, dt = \begin{cases} N & \text{si } n = m \\ 0 & \text{si } n \neq m \end{cases}$ Ec. 4.1	Reales
HAAR	$HA(n) = \int_0^1 f(t)\, HAR(n,t)\, dt$ Ec. 5.5	RECTANGULAR HAR(n,t)	$\int_0^1 HAR(m,t) \cdot HAR(n,t) = \begin{cases} 1 & \text{si } n = m \\ 0 & \text{si } n \neq m \end{cases}$ Ec. 5.2	Reales

TABLA 7.1 Definición de las Transformadas y Comparación entre ellas.

TRANSFORMADA	TRANSFORMADA INVERSA	SERIES DE TIEMPO	TIPO DE SIMETRIA DE SUS FUNCIONES
FOURIER	$f(t) = \int_{-\infty}^{\infty} F(f)\, e^{j2\pi ft}\, df$ Ec. 2.6	$f(t) = 0.5a_o + \sum_{n=1}^{\infty}\left[a_n \cos(2\pi f_o nt) + b_n \text{sen}(2\pi f_o nt)\right]$ Ec. 2.10	$\cos \omega t$ - simetría par $\text{sen}\, \omega t$ - simetría non
HARTLEY	$f(t) = \int_{-\infty}^{\infty} H(f)\, \text{cas}\,(2\pi ft)\, dt$ Ec. 3.6	Igual que Fourier	Igual que Fourier
MAXIMA ENTROPIA	Igual que Fourier	No comparable	No comparable
WALSH	$f(t) = \sum_{k=0}^{\infty} F(k) WAL(k,t)$ Ec. 4.13	$f(t) = a_o WAL(0,t) + \sum_{n=1}^{N-1} a_n WAL(n,t)$ Ec. 4.14	CAL(k,t) - Simetría Par SAL(k,t) - Simetría Non Ec. 4.2
HAAR	$f(t) = \sum_{n=0}^{\infty} C_n HAR(n,t)$ Ec. 5.6	$f(t) = c_o + \sum_{n=1}^{\infty} c_n HAR(n,t)$ Ec. 5.8	No comparable

TABLA 7.2 Transformadas Inversas, Series de Tiempo y Simetría entre sus Funciones.

TRANSFORMADA	LINEALIDAD $f_1(t) + f_2(t)$	ESCALAR $f(kt)$	DESPLAZAMIENTO EN EL TIEMPO $f(t-t_o)$	CONVOLUCION $f_1(t) * f_2(t)$
FOURIER Tabla 2.2	$F_1(f) + F_2(f)$	$K^{-1}F(t/k)$	$F(f)e^{-j2\pi f t_o}$	$F_1(f)F_2(f)$
HARTLEY Tabla 3.1	$H_1(f) + H_2(f)$	$K^{-1}H(t/k)$	$H(-f)sen(2\pi t_o f) + H(f)cos(2\pi t_o f)$	$\frac{1}{2}[H_1(f)H_2(f) - H_1(-f)H_2(-f) + H_1(f)H_2(-f) + H_1(-f)H_2(-f)]$
MAXIMA ENTROPIA	$M1(f) + M_2(f)$	Igual que Fourier	Igual que Fourier	Igual que Fourier
WALSH	$W_1(f) + W_2(f)$	No comparable	No comparable	CONVOLUCION DYADICA $R_F(\tau) = 1/N \sum_{i=0}^{N-1} x_i y_{(\tau \oplus i)}$
HAAR	$HA_1(f) + HA_2(f)$	No comparable	No comparable	----------------------

TABLA 7.3 Propiedades entre Transformadas.

TRANSFORMADAS	DEFINICION (DISCRETO)	TIPO DE IMPLANTACION COMPUTACIONAL	NUMERO DE PUNTOS EN EL ALGORITMO	CANTIDAD DE OPERACIONES
FOURIER DFT	$F(k) = N^{-1} \sum_{n=0}^{N-1} f(nT)\, e^{-2\pi jnk/N}$ Ec. 2.14	RAPIDA FFT Algoritmo Sande-Tukey (Montaño, 1989) Apéndice 2	Potencia de dos $N = 2^p$	$N(\mathrm{Log}_2 N)$
HARTLEY DHT	$H(\nu) = N^{-1} \sum_{\tau=0}^{N-1} f(\tau)\, \mathrm{cas}(2\pi\nu\tau / N)$ Ec.3.7	RAPIDA FHT Algoritmo de Bracewell (Bracewell,1986) Apéndice 3	Potencia de dos $N = 2^p$	$N(\mathrm{Log}_2 N)$
MAXIMA ENTROPIA MEM	$P(f) = \sum_{j=-m}^{m} \Phi_j \, z^{-j}$ Ec. 6.1	No es una Transformada rápida. MEM El algorimo utilizado fue el de Burg y Andersen (Press,1986) Apéndice 6	cualquier valor N	Depende del número de polos
WALSH DWT	$W(k) = {}^{1}/_{N} \sum_{i=0}^{N-1} f(nT)\, \mathrm{WAL}\,(k,i)$ 2 Ec. 4.10	RAPIDA FWT (Beauchamp,1975) Apéndice 4	Potencia de dos $N = 2^p$	$N(\mathrm{Log}_2 N)$
HAAR DHAT	$X_n = {}^{1}/_{N} \sum_{n=0}^{N-1} x_i\, \mathrm{HAR}(n, {}^{i}/_{N})$ Ec.5.9	RAPIDA FHAT (Beauchamp,1975) Apéndice 5	Potencia de dos $N = 2^p$	$2(N-1)$

TABLA 7.4 Propiedades Transformada Discreta

TRANSFORMADA	TRANSFORMADA INVERSA (DISCRETA)	VELOCIDAD (DEPENDIENDO DE LA CANTIDAD DE PUNTOS TIPO DE MAQUINA,ETC. DE MAYOR A MENOR)	RELACION ENTRE SUS FUNCIONES
FOURIER DFT	$f(nT) = \sum_{n=0}^{N-1} F(k)\, e^{2\pi jnk/N}$ Ec 2.15	#4	2coskcosp = cos(k-p) + cos(k + p) 2senkcosp = sen(k-p) + cos(k + p) 2cosksenp = -sen(k-p) + sen(k + p) 2senksenp = cos(k-p)-cos(k + p)
HARTLEY DHT	$f(\tau) = \sum_{v=0}^{N-1} H(v) cas(2\pi v\tau/N)$ Ec.3.8	#3	cas(-t) = cas'(t) = cost - sen t cas(A + B) = cos A casB + sen A cas' B cas(A-B) = cos A cas' B + sen A cas B cas A cas B = cos (A-B) + sen (A + B) cas A + cas B = 2 cas ½ (A + B) cos ½ (A-B) cas A - cas B = 2 cas' ½ (A + B) sen ½ (A-B)
MAXIMA ENTROPIA MEM	Igual que Fourier	#5	Igual que Fourier
WALSH DWT	$f(nT) = \sum_{k=0}^{\infty} W(k)\ WAL(k,t)$ Ec. 4.21	#2	WAL(k,i)WAL(p,i) = WAL[(k ⊕p),i] CAL(k,i)CAL(p,i) = CAL[(k ⊕p),i] SAL(k,i)CAL(p,i) = SAL[(p ⊕(k-1)) + 1,i] CAL(k,i)SAL(p,i) = SAL[(k ⊕p-1)) + 1,i] SAL(k,i)SAL(p,i) = CAL[(k-1) ⊕(p-1),i]
HAAR DHAT	$x_i = \sum_{n=0}^{N-1} X_n HAR(n, i/N)$ Ec.5.10	#1	HAR(0,t) } Grupo 1 HAR(1,t) } Grupo 2 HAR(2,t) ,HAR(3,t) } Grupo 3 HAR(4,t),HAR(5,t) Grupo 4 HAR(6,t),HAR(7,t) HAR(8,t),HAR(9,t) Grupo 5 HAR(10,t),HAR(11,t) HAR(12,t)HAR(13,t) HAR(14,t),HAR(15,t)

TABLA 7.5 Transformadas Inversas, Velocidad y Relación entre sus funciones

RELACION CON ⇒ TRANSFORMADAS ⇓	FOURIER	HARTLEY	MAXIMA ENTROPIA	WALSH	HAAR
FOURIER	------------------	$\frac{\lvert H(f)\rvert^2 + \lvert H(-f)^2\rvert}{2}$ Ec. 3.4	$\sqrt{P(f)}$	$F(k) = \frac{1}{N}\sum_{k=0}^{N-1} W(k)[\sum_{n=0}^{N-1} WAL(k,nT)\ e^{-2\pi nk/N}]$ Ec. 4.23	
HARTLEY		----------------------			
MAXIMA ENTROPIA			---------------		
WALSH	$W(k) = \frac{1}{N}\sum_{k=0}^{N-1} F(k)[\sum_{i=0}^{N-1} WAL(k,i)\ e^{2\pi n/N}]$ Ec. 4.22			--	
HAAR				$HAR(2^p+n,t) = \frac{1}{2\sqrt{2^p}}\sum_{k=0}^{2^{p+1}-1}[WAL(2n,\frac{k}{2^{p+1}}) - WAL(2n+1,\frac{k}{2^{p+1}})]WAL(k,t)$ Ec. 5.7	--------

Tabla 7.6 Relaciones entre Transformadas.

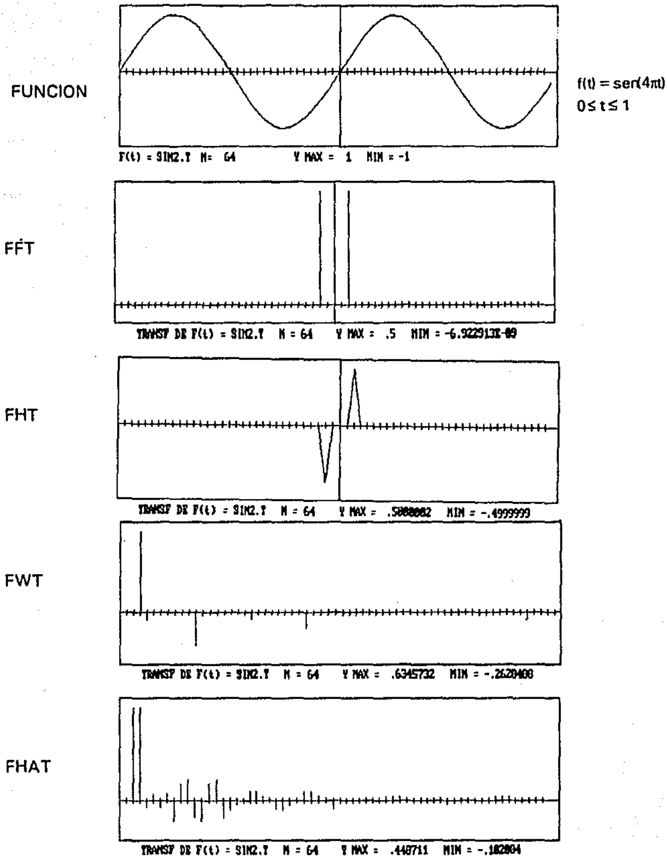


Fig. 7.1 Función Senoidal y sus Transformadas.

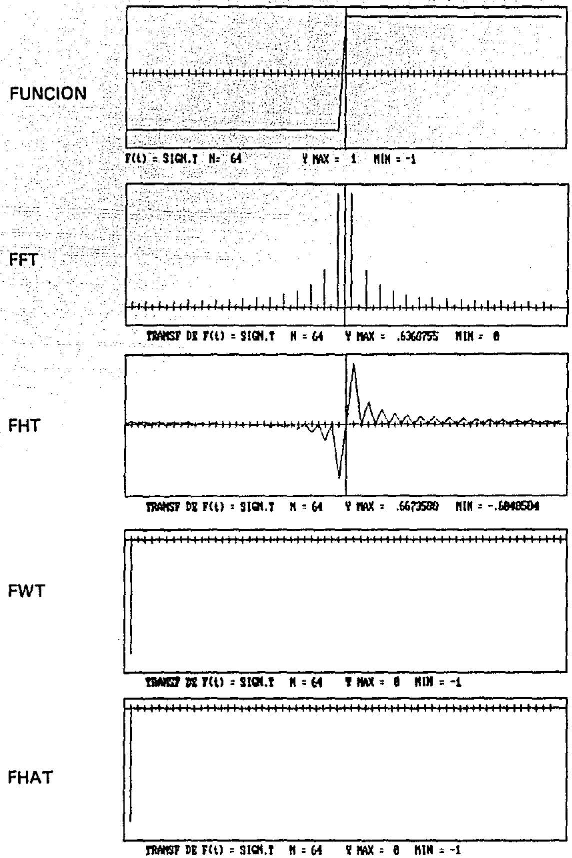


Fig. 7.2 Función Signo y sus Transformadas

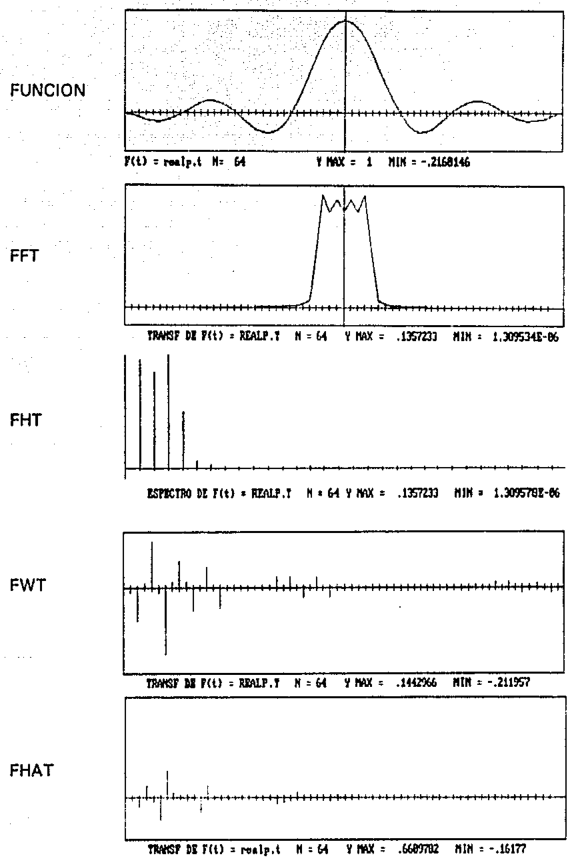


Fig. 7.3 Funcion Sa(x) y sus Transformadas.

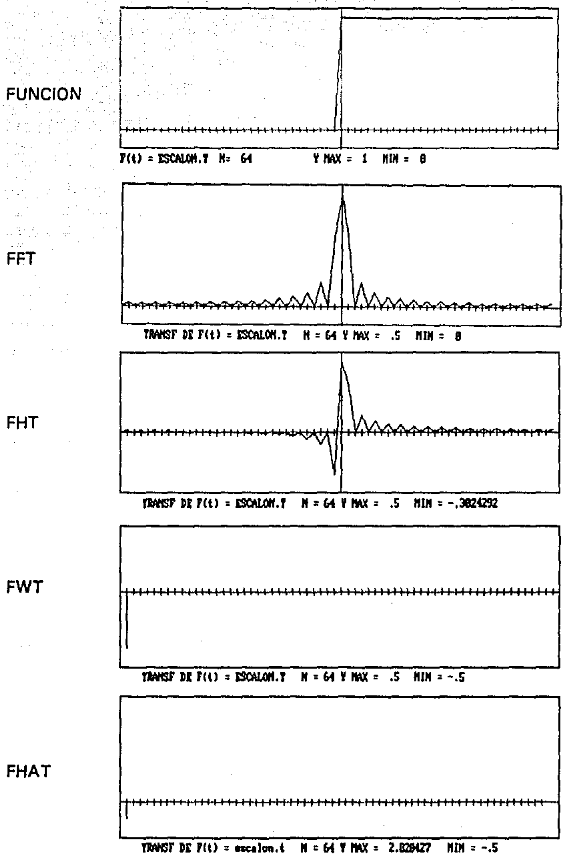


Fig. 7.4 Función Escalón y sus Transformadas.

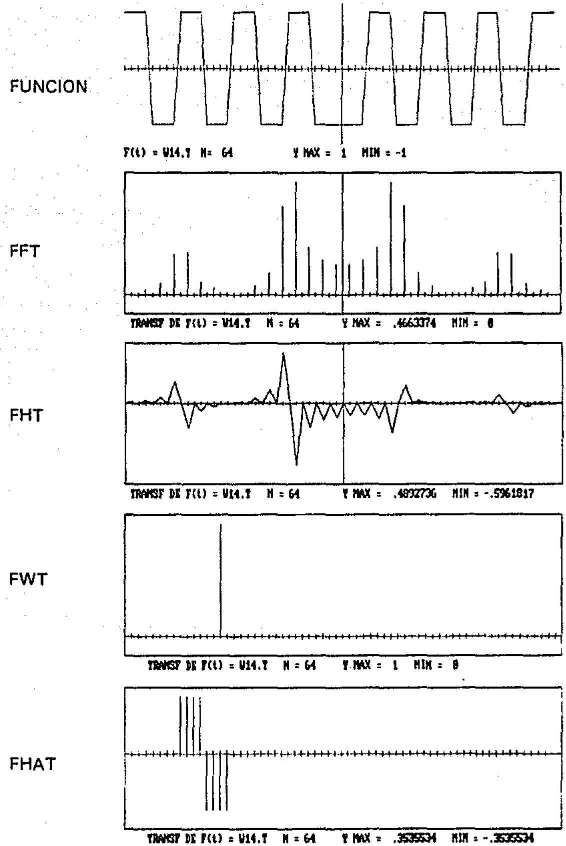


Fig. 7.5 Función 14 de Walsh y sus Transformadas.

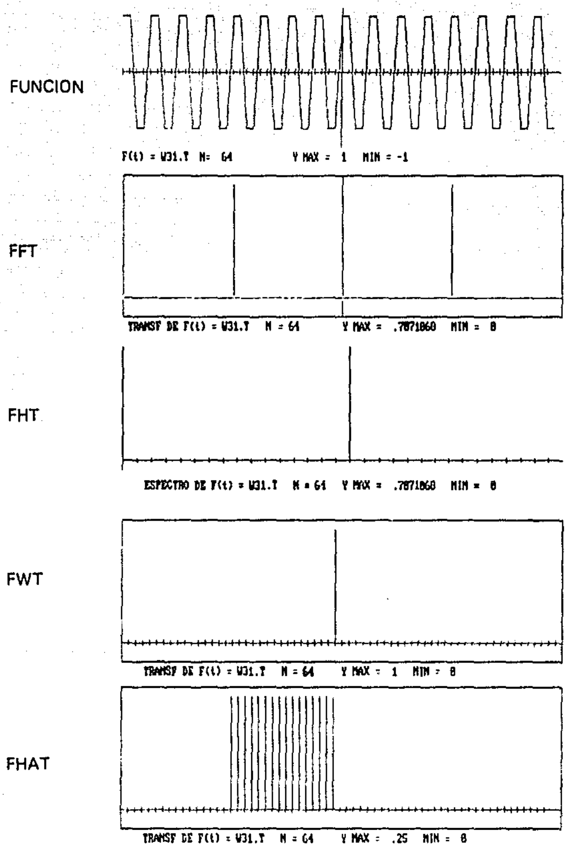


Fig. 7.6 Función 31 de Walsh y sus Transformadas.

CONCLUSIONES

NATURALEZA DE SUS FUNCIONES Y SIMETRÍA.

Las Transformadas descritas en los capítulos anteriores se pueden clasificar principalmente en 2 grupos (Tabla 7.1), dependiendo del tipo de funciones en las que se descomponen: ***Las Transformadas de Fourier, Hartley, y el Método de Máxima Entropía*** descomponen en funciones de tipo senoidal y, las de *WALSH y HAAR* en funciones de tipo rectangular. Todas estas funciones forman un grupo ortogonal y completo[1].

Estas Funciones ya sea reales o complejas tiene ciertas características de simetría, por ejemplo, funciones con simetría par (Tabla 7.2):

coswt
CAL(k,t)

y funciones con simetría non:

senwt
SAL(k,t)

Como se había mencionado las funciones CAL y SAL tienen ciertas relaciones con las funciones seno y coseno.

Cabe hacer notar que las únicas transformadas que utilizan números complejos son Fourier y Máxima Entropía, todas las demás manejan números reales.

PROPIEDADES

En cuanto a propiedades (Tabla 7.3) se tiene que todas las transformadas cumplen con las propiedad de linealidad. Por otro lado otras propiedades sólo son comunes para funciones que se descomponen en

[1] Ver Cap. I

funciones senoidales (Tabla 7.1) como son: *propiedad escalar, desplazamiento en el tiempo y convolución*. (ver convolución Dyadica). Es claro que la propiedad de desplazamiento en el tiempo, por la naturaleza de las transformadas no tiene relación entre ellas, es decir, una función coseno desfasado, cumple con propiedades (tablas 2.2 y 3.1) sólo para Fourier, Hartley y Máxima Entropía y para las otras dos, no tiene una interpretación tan directa aunque presenta un comportamiento muy particular.[2]

FORMALISMO DISCRETO Y APLICACION NUMERICA

Considerando las definiciones discretas de las Transformadas y Antitransformada se puede observar que Hartley, WALSH, HAAR (Tablas 7.4 y 7.5) puede utilizar el mismo algoritmo[3] para la transformada y antitransformada. Esta característica en el momento de evaluarse numéricamente requiere de menos recursos computacionales en cuanto a programa que el algoritmo de Fourier. El método de Máxima Entropía en este caso no se considera, porque es un método alterno a Fourier para obtener el espectro de potencia, por lo que se considerará como sección aparte.

El tiempo de cálculo como se puede observar en las Figuras a y b aumenta para cada transformada conforme se incrementa el número de puntos de la serie de tiempo, así como la diferencia en tiempo entre ellas. La Transformada más rápida resultó ser FHAT y le siguen FWT, FHT, FFT y por último el método de Máxima Entropía. Este método no se consideró en las gráficas porque el tiempo de cálculo varía con el número de polos seleccionado, pero en general siempre fue mayor que el de las otras transformadas. (También se debe considerar que este método no es una transformada rápida).

[2] Ver referencia Beauchamp, 1975 p.48 Tabla 2.2
[3] Sólo cambia la magnitud 1/N

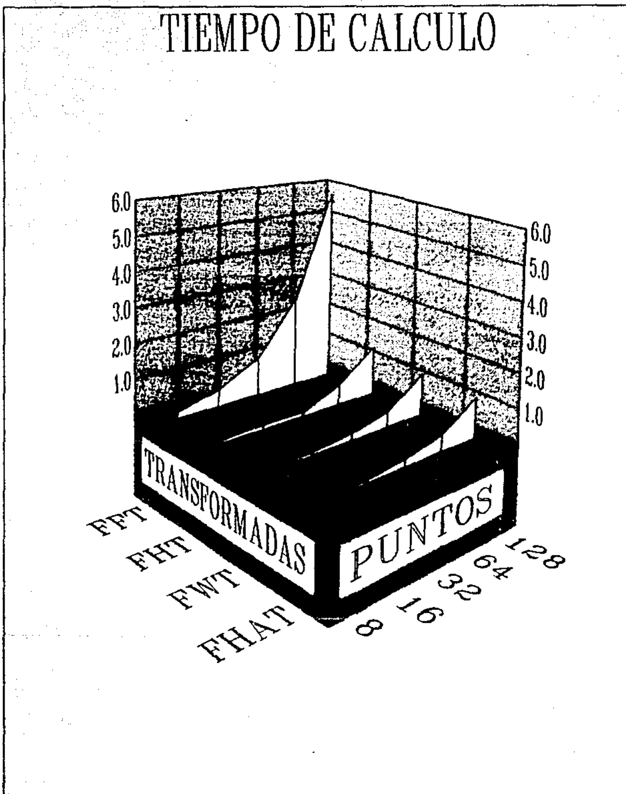


Fig. a Tiempo de Cálculo (8-128 puntos)

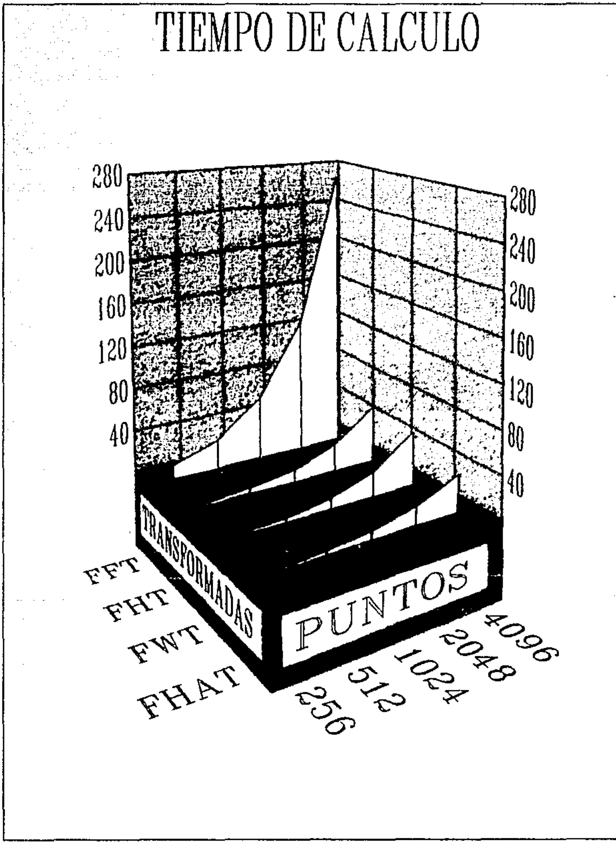


Fig. b Tiempo de Cálculo (256-4096)

Tomando en cuenta la característica mencionada anteriormente de la naturaleza de sus funciones se puede observar en las Figs. 7.* que si la función es de tipo senoidal, suave o sin discontinuidades el ancho de banda del espectro por FFT, FHT y MEM es más pequeño que FWT y FHAT, y viceversa para funciones de tipo rectangular o con discontinuidades. Dentro de las señales de tipo rectangular pueden entrar las señales digitalizadas con un valor bajo de resolución (bits de resolución) ver Figs. 7.*

Para funciones con nivel de ruido se puede observar que FFT, FHT, FWT y FHAT son sensibles a este tipo de característica.

MEM es especialmente útil por su formalismo para estos casos, por lo que no se aplicó para otro tipo de funciones (sin ruido, ver sección 6.5). Aunque el espectro es aproximado por una serie determinada por el número de polos, la relación no es lineal, esto es, el ruido no disminuye incrementado el número de polos. Por resultados de varias espectros de funciones con ruido obtenidos por MEM, se puede concluir que existe para cada función un número máximo de polos para obtener el espectro, después de este valor el espectro de la señal se distorsiona de igual manera que con un número bajo de polos[4].

En cuanto a propiedades (Tabla 7.3) debido a la naturaleza de las transformadas, la mayor parte sólo fueron aplicables para FFT, FHT, y MEM. En varias de las características no se encontró suficiente información en FHAT y FWT por lo que podrían ser objeto de estudio. (Tabla 7.6)

A lo largo del desarrollo de esta tesis y de todo aquello que implicó poder escribir estas últimas líneas, una de las conclusiones personales más importantes, fue que a pesar del desarrollo tan variado que existe en esta y otras áreas relacionadas, todavía es muy poco el uso (por lo menos en México) que se le ha dado a estas técnicas, que a la fecha no son lo último que ha salido (Cody, 1992). Esto a la única conclusión que me hace

---

[4] Ver sección 6.5

llegar, es que se requiere de un gran esfuerzo para el desarrollo de infraestructura de investigación en nuestro país....

...... espero que esta tesis, realmente sea parte de este esfuerzo.

Eduardo Gómez Ramírez

BIBLIOGRAFIA

LIBROS Y MANUALES

Beuachamp, K. G. (1975)," Walsh Function and their applications",University of Lancaster,Academic Press, Great Britain.

Bracewell,R. (1965) "The Fourier Transform and its aplications" McGraw-Hill,USA.

Bracewell,R. (1986) "The Hartley Transform", Oxford University Press,New York.

Brigham, E. (1974) "The Fast Fourier Transform", Prentice Hall, USA.

Cooper, William David. (1982) "Instrumentación Electrónica y mediciones", Prentice Hall. México.

Embree, Paul M. & Kimble, Bruce (1991) ,"C Language Algorithms for Digital Signal Processing", Prentice Hall, New Jersey.

Lathi, B.P. (1986) "Sistemas de Comunicación", Interamericana. MEXICO.

Lathi , B. P. (1987) "Introducción a la Teoría y Sistemas de Comunicación". Limusa. MEXICO.

Press, William H., Brian, Flannery P., Teukolsky, Saul A., Vetterling, William T. (1986). "Numerical Recipes", Cambridge University Press, USA.

Sohie, Guy R. L. (1991) "Implementation of Fast Fourier Transforms on Motorola's DSP56000/DSP56001 and DSP96002 Digital Signal Processors". Motorola, Inc. USA.

Stearns, Samuel D. & David, Ruth A.(1988) " Signal Processing Algorithms". Prentice Hall, New Jersey. USA

Zill, Dennis G. (1988) Ecuaciones Diferenciales con Aplicaciones. Segunda Edición. Grupo Editorial Iberoaméricana. México.

ARTICULOS Y PUBLICACIONES

Bones, Theodore M. Jr.. (1987). "Fourier Transforms". LEDS Publishing. Co. July/August. p. 40-48

Bracewell,R. (1989) "La Transformación de Fourier", Scientific American, p.56-64,Agosto.

Cody, M. (1992). "The Fast Wavelet Transform", Dr. Dobb's Journal. Abril. p. 16-28.

Del Vivar Plascencia, Rayón Villela, P., Figueroa Nazuno, J., (1990), Implementación de un programa de estimación de máxima entropía bidimensional, XXXIII Congreso Nacional de Física, 22 al 26 de octubre, Ensenada, Baja California, México.

Kay, Steven M.; Marple, Stanley L. Jr. ; (1981) " Spectrum Analysis - A Modern Perspective". Proceedings of the IEEE. Vol. 69, No. 11, p.1380-1419.November.

Loza Garay, E., Carrera Abarca, M., Franco Gutiérrez, L., & Figueroa Nazuno, J. (1990) , Implementación de la Transformada de Fourier para grandes volúmenes de datos, XXXIII Congreso Nacional de Física, 22 al 26 de octubre, Ensenada, Baja California, México

Maqusi, M. (1973)." On Moments and Walsh Characteristic Functions". IEEE Transactions on Communications." June

Montaño,J.C.,Florida,M.C.,Castilla, M., López, A., Gutiérrez, J. (1986). "Realización de la FFT en PC". Mundo Electrónico, 73-80, Diciembre.

Narayana, Siva Bala & Prabhu, K.M.M. (1991) Fast Hartley Transform Pruning. IEEE Transactions on Signal Processing. Vol. 39 No. 1 p.230-233. January.

O'Neill,M. (1988) "Faster than Fast Fourier", BYTE,p. 293-300,Abril.

Raggi, González M. G. (1992). Fundamentos e implementación de la Transformada Discreta del Coseno para el análisis y Codificación de Señales en Comunicación. "III Congreso Internacional de Electrónica y Comunicaciones UDLA'92", del 17 al 20 de febrero. Universidad de las Américas, Cholula, Puebla, México.

Robinson, E. A. (1982). " A Historical Perspective of Spectrum Stimation". Proceedings of the IEEE. Vol. 70 No. 90.p. 885-907. September.

Shore. John E. (1973) "On the Application of Haar Functions". IEEE Transactions on Communications. Vol. 21, No. 3. p.209-216,March.

Thrane,N. (1979) "The Discrete Fourier Transform and FFT Analysers", Brüel & KJÆR. Technical Review, no.1.

APENDICE 1

FUNCIONES DE RADEMACHER

Las funciones de Rademacher son funciones parecidas a las de Walsh, con valores de amplitud de 1 y -1, y a excepción de R(0,t) todas tienen simetría non. Estas funciones pueden ser derivadas de las funciones seno, que tienen la misma posición cuando cruzan el eje horizontal:

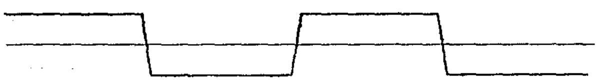
$$R(n,t) = \text{sign}(\text{sen } (2^n \pi t)) \qquad \text{Ec. A1.1}$$

donde: n es el número de función de Rademacher

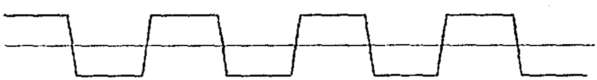
t es el intervalo de tiempo $0 \leq t \leq 1$

A continuación se presentan algunas funciones de Rademacher:

R(1,t)



R(2,t)



R(3,t)

R(4,t)

APENDICE 2

LISTADO DEL PROGRAMA FFT

```
DECLARE FUNCTION ESCRIBE! (A!)

''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''
    ' 5/3/92
    ' TRANSFORMADA RAPIDA DE FOURIER
    ' ALGORITMO SANDE-TUKEY

''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''
    ' 'DIMENSIONAR MATRICES '
    N1 = 4096
    DIM H(N1): ' //DATOS EN FUNCION DEL TIEMPO
    DIM F(N1): ' //DATOS EN FUNCION DE LA
FRECUENCIA
    DIM I(N1)
    DIM FASE1(N1)
    DIM AV(N1)
    DIM BV(N1)
    DIM BUFERE(N1)
    DIM BUFEIM(N1)
    DIM WRE#(N1)
    DIM WIM#(N1)
    DIM FASE(N1)
    '// OPCIONES DEL MENU PRINCIPAL
    DIM OPCIONES$(10)
    OPCIONES$(1) = " 1.- CARGAR DATOS f(t)"
    OPCIONES$(2) = " 2.- CARGAR DATOS f(t) DE
ARCHIVO"
    OPCIONES$(3) = " 3.- CARGAR DATOS F(f) DE
ARCHIVO"
    OPCIONES$(4) = " 4.- SALVAR
TRANSFORMADA H(F)"
    OPCIONES$(5) = " 5.- OBTENER
TRANSFORMADA DE FOURIER (FFT)"
    OPCIONES$(6) = " 6.- GRAFICAR DATOS f(t)
(TIEMPO)"
    OPCIONES$(7) = " 7.- GRAFICAR ESPECTRO"
    OPCIONES$(8) = " 8.- GRAFICAR FASE"
    OPCIONES$(9) = " 9.- DESPLEGAR DATOS F(f) "
    OPCIONES$(10) = "10.- SALIDA"
    PI = 3.141592653589795#

''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''
    ' MENU DE OPCIONES

''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''
START:
    CLS
    LOCATE 1, 27
    PRINT "TRANSFORMADA RAPIDA DE FOURIER
(FFT)"
    FOR A = 1 TO 10
        LOCATE A + 5, 25
        COLOR 1, 0
        PRINT OPCIONES$(A)
    NEXT A
    LOCATE 6, 25
    COLOR 3, 0
    PRINT OPCIONES$(1)
    POSICION = 1
    SALIDA = 0
REP:
    TECLA$ = INKEY$
    IF TECLA$ = "" THEN GOTO REP
    IF ASC(MID$(TECLA$, 1, 1)) = 0 THEN
        SELECT CASE ASC(MID$(TECLA$, 2, 1))
        CASE 72
            COLOR 1, 0
            LOCATE POSICION + 5, 25
            PRINT OPCIONES$(POSICION)
            POSICION = POSICION - 1
            IF POSICION <= 0 THEN POSICION =
10
            COLOR 3, 0
            LOCATE POSICION + 5, 25
            PRINT OPCIONES$(POSICION)
        CASE 80
            COLOR 1, 0
            LOCATE POSICION + 5, 25
            PRINT OPCIONES$(POSICION)
            POSICION = POSICION + 1
            IF POSICION > 10 THEN POSICION = 1
            COLOR 3, 0
            LOCATE POSICION + 5, 25
            PRINT OPCIONES$(POSICION)
        CASE 71
            COLOR 1, 0
            LOCATE POSICION + 5, 25
            PRINT OPCIONES$(POSICION)
            POSICION = 1
            COLOR 3, 0
            LOCATE POSICION + 5, 25
            PRINT OPCIONES$(POSICION)
        CASE 79
            COLOR 1, 0
            LOCATE POSICION + 5, 25
            PRINT OPCIONES$(POSICION)
            POSICION = 10
            COLOR 3, 0
            LOCATE POSICION + 5, 25
            PRINT OPCIONES$(POSICION)
        CASE ELSE
        END SELECT
    END IF
    IF (ASC(TECLA$) = 13) THEN
        ON POSICION GOSUB PIDE, CARGA,
CARGA1, SALVA1, 9000, GRAFICA, GRAFESP,
FASE, LISTA
        IF POSICION <> 10 THEN
            GOTO START
        ELSE
            SALIDA = 1
        END IF
    END IF
```

```
    IF (ASC(TECLA$) < > 27) AND (SALIDA = 0)
THEN GOTO REP
    END


PIDE:

'************************************************************
'     ' PEDIR DATOS
'************************************************************
   CLS
   INPUT "Numero de puntos [1..8192]"; N
   P = INT(LOG(N) / LOG(2))

'************************************************************
'     ' PEDIR DATOS DE LA TRASFORMADA
'************************************************************
   FOR I = 0 TO (N - 1)
    PRINT "F("; I; ") = "; : INPUT H(I)
   NEXT I
   GOSUB SALVA: ' //SALVAR LOS DATOS
CAPTURADOS
   RETURN


SALVA:

'************************************************************
'    'SALVAR LOS DATOS EN DOMINIO DEL TIEMPO
'************************************************************
    CLS
    LOCATE 1, 1
    INPUT "NOMBRE DEL ARCHIVO H(T) (*.T)";
NOMBRE$
    OPEN NOMBRE$ FOR OUTPUT AS #1
    PRINT #1, N
    FOR NPTS = 0 TO N - 1
          A$ = STR$(H(NPTS))
          IF MID$(A$, 1, 1) = " " AND MID$(A$, 2,
1) = "." THEN
             A$ = "0" + MID$(A$, 2, LEN(A$) - 1)
          END IF
          IF MID$(A$, 1, 1) = "-" AND MID$(A$, 2,
1) = "." THEN
             A$ = "-0" + MID$(A$, 2, LEN(A$) - 1)
          END IF
          PRINT #1, A$
    NEXT NPTS
    CLOSE #1
    RETURN


SALVA1:

'************************************************************
'    'SALVAR LOS DATOS EN DOMINIO DE LA
FRECUENCIA
'************************************************************
    CLS
    LOCATE 1, 1
    INPUT "NOMBRE DEL ARCHIVO H(F) (*.F)";
NOMBRE1$
    OPEN NOMBRE1$ FOR OUTPUT AS #1
    PRINT #1, N
    FOR NPTS = 0 TO N - 1
          A$ = STR$(F(NPTS))
          IF MID$(A$, 1, 1) = " " AND MID$(A$, 2,
1) = "." THEN
             A$ = "0" + MID$(A$, 2, LEN(A$) - 1)
          END IF
          IF MID$(A$, 1, 1) = "-" AND MID$(A$, 2,
1) = "." THEN
             A$ = "-0" + MID$(A$, 2, LEN(A$) - 1)
          END IF
          PRINT #1, A$
    NEXT NPTS
    CLOSE #1
    RETURN


CARGA:

'************************************************************
'    ' CARGAR PUNTOS EN DOMINIO DEL TIEMPO
'************************************************************
    CLS
    LOCATE 1, 1
    INPUT "NOMBRE DEL ARCHIVO H(T) [*.T]";
NOMBRE$
    OPEN NOMBRE$ FOR INPUT AS #1
    INPUT #1, NPT$
    N = VAL(NPT$)
    FOR NPTS = 0 TO N - 1
         INPUT #1, CADENA$
         H(NPTS) = VAL(CADENA$)
    NEXT NPTS
    CLOSE #1
    ' // OBTENER ( POWER INDEX )
    P = INT(LOG(N) / LOG(2))
    RETURN

CARGA1:

'************************************************************
'    'CARGAR DATOS EN DOMINIO DE LA
FRECUENCIA
'************************************************************
    CLS
    LOCATE 1, 1
    INPUT "NOMBRE DEL ARCHIVO H(F) [*.F]";
NOMBRE$
    OPEN NOMBRE$ FOR INPUT AS #1
    INPUT #1, NPT$
    N = VAL(NPT$)
    FOR NPTS = 0 TO N - 1
```

```
        INPUT #1, CADENA$
        F(NPTS) = VAL(CADENA$)
    NEXT NPTS
    CLOSE #1
    ' // OBTENER POWER INDEX
    P = INT(LOG(N) / LOG(2))
    RETURN

GRAFICA:

'**************************************
*************************
   'GRAFICAR LOS DATOS

'**************************************
*************************
   CLS

'**************************************
*************************
   ' OBTENER EL MAXIMO Y EL MINIMO

'**************************************
*************************
   MAYOR = H(0)
   MENOR = H(0)
   FOR k = 1 TO N - 1
    IF H(k) > MAYOR THEN MAYOR = H(k)
    IF H(k) < MENOR THEN MENOR = H(k)
   NEXT k
   '***** AJUSTAR LIMITES PARA GRAFICAR
******
   DY = MAYOR - MENOR
   IF DY = 0 THEN DY = MAYOR
   DY1 = (MAYOR * 320) / DY
   DX = 640 / N
   '***** GRAFICAR ****
   SCREEN 9
   ' //DESPLEGAR DATOS DE LA GRAFICA
   LOCATE 25, 5
   PRINT "F(t) = "; NOMBRE$; "   PUNTOS = "; N;
"    Y MAX = "; MAYOR; "   MIN = "; MENOR
   ' // DIBUJAR EL MARCO DE LA GRAFICA
   LINE (0, 0)-(639, 349), 14, B
   LINE (N / 2 * DX, 1)-(N / 2 * DX, 400), 14
   ' // DIBUJAR LINEA CON Y=0
   LINE (1, DY1 + 10)-(640, DY1 + 10), 14
   '*** GRAFICAR LOS PUNTOS **
   FOR I = 0 TO N - 1
     Y = (H(I) * 320) / DY: Y1 = (H(I + 1) * 320) /
DY
     Y = (DY1 - Y)
     Y1 = (DY1 - Y1)
     LINE (I * DX, Y + 10)-((I + 1) * DX, Y1 + 10),
3
     LINE (I * DX, DY1 + 13)-(I * DX, DY1 + 7), 4
   NEXT I
LB2: A$ = INKEY$: IF A$ = "" THEN GOTO LB2
   RETURN

READA:

'**************************************
*************
   '   REACOMODAR DATOS

'**************************************
*************
    FOR I = 0 TO N / 2 - 1
     T = H(I + N / 2)
     H(I + N / 2) = H(I)
     H(I) = T
    NEXT I
    RETURN

READAF:

'**************************************
*************
   ' REORDENAR DATOS PARA GRAFICAR

'**************************************
*************
   FOR I = 0 TO N / 2 - 1
    T = F(I + N / 2)
    F(I + N / 2) = F(I)
    F(I) = T
   NEXT I
   RETURN

GRAFESP:

'**************************************
*************
   ' REDONDEO DE LOS DATOS

'**************************************
*************
   CLS
   PRINT "NUMERO DE PUNTOS = "; N
   INPUT "LIMITE MINIMO "; XMIN
   INPUT "LIMITE MAXIMO "; XMAX
   INPUT "TIPO DE GRAFICA (1/CONTINUA,
ENTER/DISCRETA)"; GRA
   IF XMAX = 0 THEN
     XMIN = 0
     XMAX = N - 1
   END IF


'**************************************
*************
   'OBTENER VALORES MAXIMO Y MINIMO

'**************************************
*************
   MAYOR = F(XMIN)
   MENOR = F(XMAX)
   FOR k = XMIN + 1 TO XMAX - 1
    IF F(k) > MAYOR THEN MAYOR = F(k)
    IF F(k) < MENOR THEN MENOR = F(k)
   NEXT k
   CLS
   SCREEN 9
   LOCATE 25, 5
   PRINT "TRANSF DE F(t) = "; NOMBRE$; "   N = ";
N
   LOCATE 25, 41
   PRINT "Y MAX = "; MAYOR; "  MIN = "; MENOR

'**************************************
***********
```

```
   ' OBTENER LAS RELACIONES NECESARIAS PARA
GRAFICAR

'***************************************
***********
   DY = MAYOR - MENOR
   DY1 = (MAYOR * 320) / DY
   DX = 640 / (XMAX + 1 - XMIN)
   '// DIBUJAR MARCO DE LA GRAFICA
   LINE (0, 0)-(639, 349), 14, B
   LINE ((N / 2 - XMIN) * DX, 1)-((N / 2 - XMIN) * DX,
400), 14
   '// DIBUJAR LINEA Y=0
   LINE (1, DY1 + 10)-(640, DY1 + 10), 14
   '// GRAFICAR LOS PUNTOS
   FOR I = XMIN TO XMAX
      Y = (F(I) * 320) / DY: Y1 = (F(I + 1) * 320) /
DY
      Y = (DY1 - Y): Y1 = (DY1 - Y1)
      '***LINEA OPCIONAL ****
      IF GRA = 1 THEN
      LINE ((I - XMIN) * DX, Y + 10)-(((I - XMIN) + 1)
* DX, Y1 + 10), 3: 'ESPECTRO CONTINUO
      END IF
      '*********************
      IF GRA <> 1 THEN
      LINE ((I - XMIN) * DX, DY1 + 10)-((I - XMIN) *
DX, Y + 10), 3: 'ESPECTRO DISCRETO
      END IF
      LINE ((I - XMIN) * DX, DY1 + 13)-((I - XMIN) *
DX, DY1 + 7), 4
   NEXT I
LB4: A$ = INKEY$: IF A$ = "" THEN GOTO LB4
   RETURN

9000 : CLS
      SCREEN 9
      PRINT "CALCULANDO"
      'SUBRUTINA PARA CALCULAR LA
TRANSFORMADA  RAPIDA DE FOURIER
   AG# = PI * 2 / N
   FOR I = 0 TO N - 1
    F(I) = H(I)
   NEXT I
   GOSUB READAF
   TIEMPO1 = TIMER 'Tiempo inicial
   FOR Z = 0 TO N / 2 - 1
    WRE#(Z) = COS(AG# * (-Z))
    WIM#(Z) = -SIN(AG# * (-Z))
   NEXT Z

'***************************************
************************
   'Calculo de la FFT

'***************************************
************************
   FOR I = 1 TO P
         L = 0: H = 0: G = N / (2 ^ I)
         FOR k = 0 TO N - 1 STEP G
           TFI = 0
           TFIFLAG = (-1) ^ (L + 1)
           FOR J = 0 TO G - 1
             TFI = J * 2 ^ (I - 1)
             R = k + J
             S = J + H
             T = J + G + H
             IF TFIFLAG > 0 THEN
                   TEMPRE = F(S) - F(T)
                   TEMPIM = I(S) - I(T)
                   BUFERE(R) = TEMPRE *
WRE#(TFI) - TEMPIM * WIM#(TFI)
                   BUFEIM(R) = TEMPRE *
WIM#(TFI) + TEMPIM * WRE#(TFI)
             ELSE
                   BUFERE(R) = F(S) + F(T)
                   BUFEIM(R) = I(S) + I(T)
             END IF
           NEXT J
           L = L + 1
           H = INT(L / 2) * G * 2
         NEXT k
         FOR II = 0 TO N - 1
           F(II) = BUFERE(II)
           I(II) = BUFEIM(II)
         NEXT II
   NEXT I
   FOR I = 0 TO N - 1
         F(I) = F(I) / N
         I(I) = I(I) / N
   NEXT I
   ' Reordenacion de bits
   FOR I = 0 TO N - 1
         INDEX% = I
         IOUT% = 0
         FOR J = 1 TO P
           TEMP% = 1 AND INDEX%
           IOUT% = IOUT% * 2
           IOUT% = IOUT% + TEMP%
           INDEX% = INDEX% \ 2
         NEXT J
         BUFERE(I) = F(IOUT%)
         BUFEIM(I) = I(IOUT%)
   NEXT I
   'Calculo de modulos y fases
   FOR I = 0 TO N - 1
     BUFERE(N) = BUFERE(0)
     BUFEIM(N) = BUFEIM(0)
     AV(I) = (BUFERE(I) + BUFERE(N - I)) / 2
     BV(I) = (BUFEIM(I) - BUFEIM(N - I)) / 2
   NEXT I
   F(0) = AV(0)
   FOR J = 1 TO N - 1
     F(J) = SQR((BV(J) ^ 2 + AV(J) ^ 2)): 'Modulo j
     IF AV(J) = 0 THEN
           IF BV(J) < 0 THEN FASE(J) = PI / 2
           IF BV(J) > 0 THEN FASE(J) = 3 * PI / 2
     END IF
     IF AV(J) <> 0 THEN FASE(J) = ATN(-BV(J) /
AV(J))
     FASE(J) = 360 * FASE(J) / 2 / PI
   NEXT J
   TIEMPO2 = TIMER
   ' //TIEMPO FINAL
   dura = TIEMPO2 - TIEMPO1
   '//REORDENAR DATOS
   GOSUB READAF
   PRINT " tiempo -->", dura
   PRINT "Presione cualquier tecla"
LB1: A$ = INKEY$: IF A$ = "" THEN GOTO LB1
RETURN

FASE:
```

```
'...........................................
.............
   'SUBRUTINA PARA GRAFICAR ESPECTRO DE
FASE

'...........................................
.............

'...........................................
............
   ' REDONDEO DE LOS DATOS

'...........................................
............
   CLS
   PRINT "NUMERO DE PUNTOS = "; N
   INPUT "LIMITE MINIMO "; XMIN
   INPUT "LIMITE MAXIMO "; XMAX
   FOR I = 0 TO (N / 2 - 1)
    T = FASE(I + N / 2)
    FASE(I + N / 2) = FASE(I)
    FASE(I) = T
   NEXT I
   CLS
   SCREEN 9

'...........................................
...........
   'OBTENER VALORES MAXIMO Y MINIMO

'...........................................
...........
   MAYOR = FASE(XMIN)
   MENOR = FASE(XMAX)
   FOR k = XMIN + 1 TO XMAX - 1
    IF FASE(k) > MAYOR THEN MAYOR = FASE(k)
    IF FASE(k) < MENOR THEN MENOR = FASE(k)
   NEXT k
   LOCATE 25, 5
   PRINT "TRANSF DE F(t) = "; NOMBRE$; "   N
=";N
   LOCATE 25, 41
   PRINT "Y MAX = "; MAYOR; "  MIN = "; MENOR

'...........................................
..........
   ' OBTENER LAS RELACIONES NECESARIAS PARA
GRAFICAR

'...........................................
...........
   DY = MAYOR - MENOR: DY1 = (MAYOR * 320)
/ DY: DX = 635 / (XMAX + 1 - XMIN)
   LINE (0, 0)-(639, 349), 14, B
   LINE ((N / 2 - XMIN) * DX, 1)-((N / 2 - XMIN) *
DX, 400), 14
   LINE (1, DY1 + 10)-(640, DY1 + 10), 14
   FOR I = XMIN TO XMAX - 1
    Y = (FASE(I) * 320) / DY: Y1 = (FASE(I + 1) *
320) / DY
    Y = (DY1 - Y): Y1 = (DY1 - Y1)
    LINE ((I - XMIN) * DX, DY1 + 10)-((I - XMIN) *
DX, Y + 10), 3
    LINE ((I - XMIN) * DX, DY1 + 13)-((I - XMIN) *
DX, DY1 + 7), 4
   NEXT I
LB7: A$ = INKEY$: IF A$ = "" THEN GOTO LB7
   FOR I = 0 TO N / 2 - 1
    T = FASE(I + N / 2)
    FASE(I + N / 2) = FASE(I)
    FASE(I) = T
   NEXT I
   RETURN


LISTA:

'...........................................
...........
   'OBTENER VALORES MAXIMO Y MINIMO

'...........................................
...........
   MAYOR = F(0)
   MENOR = F(0)
   FOR k = 1 TO N - 2
    IF F(k) > MAYOR THEN MAYOR = F(k)
    IF F(k) < MENOR THEN MENOR = F(k)
   NEXT k
   CLS
   FOR I = 0 TO N - 1
    PRINT "DATO "; I; " --> "; F(I)
    IF (I / 20) = INT(I / 20) AND I <> 0 THEN
        LOCATE 23, 10
        PRINT "<PRESIONA CUALQUIER TECLA>"
ET1:    WTEC$ = INKEY$: IF WTEC$ = "" THEN
GOTO ET1
        CLS
    END IF
   NEXT I
   PRINT
   PRINT "          VALOR MAYOR = "; MAYOR
   PRINT "          VALOR MENOR = "; MENOR
   LOCATE 23, 10
   PRINT "<PRESIONA CUALQUIER TECLA>"
LB9: A$ = INKEY$: IF A$ = "" THEN GOTO LB9
   RETURN

FUNCTION ESCRIBE (A)


END FUNCTION
```

APENDICE 3

LISTADO DEL PROGRAMA FHT

```
'*********************************************************************
' 05/03/91
'TRANSFORMADA RAPIDA DE HARTLEY

'*********************************************************************
'dimensionar matrices
'*********************************************************************
N1 = 4096
DIM F(N1) '//VECTOR PARA DATOS EN FUNCION DE LA FRECUENCIA
DIM H(N1) '//VECTOR PARA DATOS EN FUNCION DEL TIEMPO
DIM P(N1)
DIM S9(N1 / 2)
DIM T9(N1 / 2)
DIM M9(20)
DIM A9(64)
DIM V9(10)
DIM C9(10)
DIM OPCIONES$(20)
OPCIONES$(1) = " 1.- CARGAR DATOS f(t)"
OPCIONES$(2) = " 2.- CARGAR DATOS f(t) DE ARCHIVO"
OPCIONES$(3) = " 3.- CARGAR DATOS H(f) DE ARCHIVO"
OPCIONES$(4) = " 4.- CARGAR DATOS P(f) DE ARCHIVO"
OPCIONES$(5) = " 5.- OBTENER TRANSFORMADA DE HARTLEY"
OPCIONES$(6) = " 6.- OBTENER ESPECTRO DE POTENCIA"
OPCIONES$(7) = " 7.- SALVAR TRANSFORMADA DE HARTLEY "
OPCIONES$(8) = " 8.- SALVAR ESPECTRO DE POTENCIA P(f) "
OPCIONES$(9) = " 9.- GRAFICAR DATOS f(t)"
OPCIONES$(10) = "10.- GRAFICAR TRANSFORMADA DE HARTLEY H(f)"
OPCIONES$(11) = "11.- GRAFICAR ESPECTRO DE POTENCIA P(f)"
OPCIONES$(12) = "12.- LISTAR DATOS H(f)"
OPCIONES$(13) = "13.- SALIDA"
pi = 3.141592653589795#

'*********************************************************************
' MENU DE OPCIONES
'*********************************************************************
START:
CLS
LOCATE 1, 27
PRINT "TRANSFORMADA DE HARTLEY"
FOR A = 1 TO 13
 LOCATE A + 5, 25
 COLOR 1, 0
 PRINT OPCIONES$(A)
NEXT A
LOCATE 6, 25
COLOR 3, 0
PRINT OPCIONES$(1)
POSICION = 1
SALIDA = 0
REP:
 TECLA$ = INKEY$
 IF TECLA$ = "" THEN GOTO REP
 IF ASC(MID$(TECLA$, 1, 1)) = 0 THEN
  SELECT CASE ASC(MID$(TECLA$, 2, 1))
     CASE 72
      COLOR 1, 0
      LOCATE POSICION + 5, 25
      PRINT OPCIONES$(POSICION)
      POSICION = POSICION - 1
     IF POSICION < = 0 THEN POSICION = 13
      COLOR 3, 0
      LOCATE POSICION + 5, 25
      PRINT OPCIONES$(POSICION)
     CASE 80
      COLOR 1, 0
      LOCATE POSICION + 5, 25
      PRINT OPCIONES$(POSICION)
      POSICION = POSICION + 1
      IF POSICION > 13 THEN POSICION = 1
      COLOR 3, 0
      LOCATE POSICION + 5, 25
      PRINT OPCIONES$(POSICION)
     CASE 71
      COLOR 1, 0
      LOCATE POSICION + 5, 25
      PRINT OPCIONES$(POSICION)
      POSICION = 1
      COLOR 3, 0
      LOCATE POSICION + 5, 25
      PRINT OPCIONES$(POSICION)
     CASE 79
      COLOR 1, 0
      LOCATE POSICION + 5, 25
      PRINT OPCIONES$(POSICION)
      POSICION = 13
      COLOR 3, 0
      LOCATE POSICION + 5, 25
      PRINT OPCIONES$(POSICION)
     CASE ELSE
  END SELECT
 END IF
 IF (ASC(TECLA$) = 13) THEN
```

```
    ON POSICION GOSUB PIDE, CARGA, CARGA1,
CARGA2, 9000, SPECTRUM, SALVA1, SALVA2,
GRAFICA, GRAFESP, GSPECTRUM, LISTA

    IF POSICION < > 13 THEN
        GOTO START
    ELSE
        SALIDA = 1
    END IF
   END IF
   IF (ASC(TECLA$) < > 27) AND (SALIDA = 0)
THEN GOTO REP
   END


PIDE:

'***************************************
'*****************
   ' PEDIR DATOS

'***************************************
'*****************
   CLS
   INPUT "Numero de puntos [1..8192]"; N
   P = INT(LOG(N) / LOG(2))

'***************************************
'***************************
   ' PEDIR DATOS DE LA TRASFORMADA

'***************************************
'***************************
   FOR I = 0 TO (N - 1)
       PRINT "F("; I; ") = "; : INPUT H(I)
   NEXT I
   '//GUARDAR EN UN ARCHIVO LOS DATOS
CAPTURADOS
   GOSUB SALVA
   RETURN


SALVA:

'***************************************
'*************************
   'SALVAR LOS DATOS EN DOMINIO DEL TIEMPO

'***************************************
'*************************
   CLS
   LOCATE 1, 1
   INPUT "NOMBRE DEL ARCHIVO H(T) (*.T)";
NOMBRE$
   OPEN NOMBRE$ FOR OUTPUT AS #1
   PRINT #1, N
   FOR NPTS = 0 TO N - 1
       A$ = STR$(H(NPTS))
       IF MID$(A$, 1, 1) = " " AND MID$(A$, 2,
1) = "." THEN
          A$ = "0" + MID$(A$, 2, LEN(A$) - 1)
       END IF
       IF MID$(A$, 1, 1) = "-" AND MID$(A$, 2,
1) = "." THEN
          A$ = "-0" + MID$(A$, 2, LEN(A$) - 1)
       END IF
       PRINT #1, A$
   NEXT NPTS
   CLOSE #1
   RETURN

SALVA1:

'***************************************
'*************************
   'SALVAR LOS DATOS EN DOMINIO DE LA
FRECUENCIA

'***************************************
'*************************
   CLS
   LOCATE 1, 1
   INPUT "NOMBRE DEL ARCHIVO H(F) (*.H)";
NOMBRE1$
   OPEN NOMBRE1$ FOR OUTPUT AS #1
   PRINT #1, N
   FOR NPTS = 0 TO N - 1
    A$ = STR$(F(NPTS))
    IF MID$(A$, 1, 1) = " " AND MID$(A$, 2, 1) =
"." THEN
          A$ = "0" + MID$(A$, 2, LEN(A$) - 1)
    END IF
    IF MID$(A$, 1, 1) = "-" AND MID$(A$, 2, 1) =
"." THEN
          A$ = "-0" + MID$(A$, 2, LEN(A$) - 1)
    END IF
    PRINT #1, A$
   NEXT NPTS
   CLOSE #1
   RETURN

SALVA2:

'***************************************
'*************************
   'SALVAR ESPECTRO DE POTENCIA

'***************************************
'*************************
   CLS
   LOCATE 1, 1
   INPUT "NOMBRE DEL ARCHIVO P(N) (*.P)";
NOMBRE1$
   OPEN NOMBRE1$ FOR OUTPUT AS #1
   PRINT #1, NN
   FOR NPTS = 0 TO NN
    A$ = STR$(P(NPTS))
    IF MID$(A$, 1, 1) = " " AND MID$(A$, 2, 1) =
"." THEN
          A$ = "0" + MID$(A$, 2, LEN(A$) - 1)
    END IF
    IF MID$(A$, 1, 1) = "-" AND MID$(A$, 2, 1) =
"." THEN
          A$ = "-0" + MID$(A$, 2, LEN(A$) - 1)
    END IF
    PRINT #1, A$
   NEXT NPTS
   CLOSE #1
   RETURN

CARGA:

'***************************************
'***************************
```

```
    ' CARGAR PUNTOS EN DOMINIO DEL TIEMPO

'..........................................
............................
    CLS
    LOCATE 1, 1
    INPUT "NOMBRE DEL ARCHIVO H(T) [*.T]";
NOMBRE$
    OPEN NOMBRE$ FOR INPUT AS #1
    INPUT #1, NPT$
    N = VAL(NPT$)
    FOR NPTS = 0 TO N - 1
     INPUT #1, CADENA$
     H(NPTS) = VAL(CADENA$)
    NEXT NPTS
    CLOSE #1
    '//OBTENER (POWER INDEX)
    P = INT(LOG(N) / LOG(2))
    RETURN


CARGA1:

'..........................................
............................
    'CARGAR DATOS EN DOMINIO DE LA
FRECUENCIA

'..........................................
............................
    CLS
    LOCATE 1, 1
    INPUT "NOMBRE DEL ARCHIVO H(F) [*.H]";
NOMBRE$
    OPEN NOMBRE$ FOR INPUT AS #1
    INPUT #1, NPT$
    N = VAL(NPT$)
    FOR NPTS = 0 TO N - 1
     INPUT #1, CADENA$
     F(NPTS) = VAL(CADENA$)
    NEXT NPTS
    CLOSE #1
    '//OBTENER (POWER INDEX)
    P = INT(LOG(N) / LOG(2))
    RETURN

CARGA2:

'..........................................
............................
    'CARGAR DATOS DEL ESPECTRO DE POTENCIA

'..........................................
............................
    CLS
    LOCATE 1, 1
    INPUT "NOMBRE DEL ARCHIVO P(N) [*.P]";
NOMBRE$
    OPEN NOMBRE$ FOR INPUT AS #1
    INPUT #1, NPT$
    NN = VAL(NPT$)
    FOR NPTS = 0 TO NN - 1
     INPUT #1, CADENA$
     P(NPTS) = VAL(CADENA$)
    NEXT NPTS
    CLOSE #1
    RETURN

GRAFICA:

'..........................................
..........................
    'GRAFICAR LOS DATOS

'..........................................
..........................
    CLS

'..........................................
..........................
    ' OBTENER EL MAXIMO Y EL MINIMO

'..........................................
..........................
    MAYOR = H(0)
    MENOR = H(0)
    FOR K = 1 TO N - 1
     IF H(K) > MAYOR THEN MAYOR = H(K)
     IF H(K) < MENOR THEN MENOR = H(K)
    NEXT K
    '***** AJUSTAR LIMITES PARA GRAFICAR
*******
    DY = MAYOR - MENOR
    IF DY = 0 THEN DY = MAYOR
    DY1 = (MAYOR * 320) / DY
    DX = 640 / N
    '***** GRAFICAR ****
    SCREEN 9
    LOCATE 25, 5
    '***********//DESPLEGAR DATOS DE LA
GRAFICA
    PRINT "F(t) = "; NOMBRE$; "    PUNTOS = "; N;
"     Y MAX = "; MAYOR; "   MIN = "; MENOR
    '*********//HACER MARCO PARA LA
GRAFICA
    LINE (0, 0)-(639, 349), 14, B
    LINE (N / 2 * DX, 1)-(N / 2 * DX, 400), 14
    '********GRAFICAR LINEA Y = 0
    LINE (1, DY1 + 10)-(640, DY1 + 10), 14
    FOR I = 0 TO N - 1
     y = (H(I) * 320) / DY: Y1 = (H(I + 1) * 320) /
DY
     y = (DY1 - y): Y1 = (DY1 - Y1)
     LINE (I * DX, y + 10)-((I + 1) * DX, Y1 + 10),
3
     LINE (I * DX, DY1 + 13)-(I * DX, DY1 + 7), 4
    NEXT I
LB2: A$ = INKEY$: IF A$ = "" THEN GOTO LB2
    RETURN

READA:

'..........................................
.............
    '   REACOMODAR DATOS

'..........................................
.............
    FOR I = 0 TO N / 2 - 1
     T = H(I + N / 2)
     H(I + N / 2) = H(I)
     H(I) = T
```

```
    NEXT I
    RETURN

READAF:

'**************************************************
'
    ' REORDENAR DATOS PARA GRAFICAR

'**************************************************
'
    FOR I = 0 TO N / 2 - 1
     T = F(I + N / 2)
     F(I + N / 2) = F(I)
     F(I) = T
    NEXT I
    RETURN

GRAFESP:


'**************************************************
'
    ' REDONDEO DE LOS DATOS

'**************************************************
'
    GOSUB READAF
    CLS
    PRINT "NUMERO DE PUNTOS = "; N
    INPUT "LIMITE MINIMO "; XMIN
    INPUT "LIMITE MAXIMO "; XMAX
    'FOR I = 0 TO (N - 1)
    '>>>>>>>> F(I) = INT(.5 + 1000 / N *
F(I)) / 1000
    'NEXT I

    CLS
    SCREEN 9

'**************************************************
'
    'OBTENER VALORES MAXIMO Y MINIMO

'**************************************************
'
    MAYOR = F(XMIN)
    MENOR = F(XMAX)
    FOR K = XMIN + 1 TO XMAX - 1
     IF F(K) > MAYOR THEN MAYOR = F(K)
     IF F(K) < MENOR THEN MENOR = F(K)
    NEXT K
    LOCATE 25, 5
    PRINT "TRANSF DE F(I) = "; NOMBRE$; "   N
="; N
    LOCATE 25, 41
    PRINT "Y MAX = "; MAYOR; " MIN = ";
MENOR

'**************************************************
'
    ' OBTENER LAS RELACIONES NECESARIAS
PARA GRAFICAR

'**************************************************
'
    DY = MAYOR - MENOR
    DY1 = (MAYOR * 320) / DY
    DX = 640 / (XMAX + 1 - XMIN)
    '**************//GRAFICAR MARCO PARA
LA GRAFICA
    LINE (0, 0)-(639, 349), 14, B
    LINE ((N / 2 - XMIN) * DX, 1)-((N / 2 - XMIN) *
DX, 400), 14
    '*************//DIBUJAR EJE LA ABCISAS
    LINE (1, DY1 + 10)-(640, DY1 + 10), 14
    '*************//GRAFICAR LOS PUNTOS
    FOR I = XMIN TO XMAX - 1
     y = (F(I) * 320) / DY: Y1 = (F(I + 1) * 320) /
DY
     y = (DY1 - y): Y1 = (DY1 - Y1)
     LINE ((I - XMIN) * DX, y + 10)-(((I - XMIN) + 1)
* DX, Y1 + 10), 3
     LINE ((I - XMIN) * DX, DY1 + 13)-((I - XMIN) *
DX, DY1 + 7), 4
    NEXT I
LB4: A$ = INKEY$: IF A$ = "" THEN GOTO LB4
    GOSUB READAF
    RETURN

9000 :
    '                    FHTSUB
    ' This subroutine takes input f() and returns the
DHT in the same f()
    FOR I = 0 TO N - 1
         F(I) = H(I)
    NEXT I
    'REACOMODAR DATOS PARA LA
TRANSFORMADA
    GOSUB READAF
    T0 = TIMER
    CLS
    PRINT "CALCULANDO"
9030 IF P = 1 THEN
         J = F(0) + F(1)
         F(1) = F(0) - F(1)
         F(0) = J
         RETURN
    END IF
    N9 = 2 ^ (P - 2)
    NP = 4 * N9
    C9(5) = NP - 1
    C9(6) = P - 1
    IF NP = N0 THEN GOTO 9400
'Skip pretabulation
    I = 1
    M9(0) = 1
    M9(1) = 2
9202 M9(I + 1) = M9(I) + M9(I)
    I = I + 1
    IF I < P THEN GOTO 9202
    IF NP = 2 THEN GOTO 9411
'Special case
    IF NP < 8 THEN GOTO 9400
'Skip trigonometric functions
    S9(N9) = 1
    IF NP = 8 THEN S9(1) = SIN(pi / 4): GOTO
9330                          'Skip sines

'**************************************************
'
9300 'GET SINES
```

```
'*************************************
'**************
    FOR I = 1 TO 3
        S9(I * N9 / 4) = SIN(I * pi / 8)
    NEXT I                      'Coarse seed table
for sines
    H9 = 1 / 2 / COS(pi / 16)       'Initial half
secant
    '********  Fill sine table  *****
    C9(4) = P - 4
    FOR I = 1 TO (P - 4)
        C9(4) = C9(4) - 1
        V9(0) = 0
        FOR J = M9(C9(4)) TO (N9 - M9(C9(4)))
STEP M9(C9(4) + 1)
            V9(1) = J + M9(C9(4))
            S9(J) = H9 * (S9(V9(1)) + V9(0))
            V9(0) = S9(V9(1))
        NEXT J
        H9 = 1 / SQR(2 + 1 / H9):
'Half secant recursion
    NEXT I
9330 '*******GET TANGENTS*******
    C9(0) = N9 - 1
    FOR I = 1 TO (N9 - 1)
        T9(I) = (1 - S9(C9(0))) / S9(I)
        C9(0) = C9(0) - 1
    NEXT I
    T9(N9) = 1
9400
'*************************************
'***************************
    'FAST PERMUTE

'*************************************
'***************************
    '*****For P = 2, 3 permute directly*****
    IF P = 2 THEN
        V9(9) = F(1)
        F(1) = F(2)
        F(2) = V9(9)
        GOTO 9500
    END IF
    IF P = 3 THEN
        V9(9) = F(1)
        F(1) = F(4)
        F(4) = V9(9)
        V9(9) = F(3)
        F(3) = F(6)
        F(6) = V9(9)
    END IF
    IF P = 3 THEN GOTO 9500
    '******* For P = 4,5,6 (Q9 = 2,3 ), skip
structure table*****
    Q9 = INT(P / 2)
    C9(2) = M9(Q9)
    Q9 = Q9 + P MOD 2
    IF Q9 = 2 THEN
        A9(1) = 2
        A9(2) = 1
        A9(3) = 3
        GOTO 9420
    END IF
   IF Q9 = 3 THEN
    A9(1) = 4
    A9(2) = 2
    A9(3) = 6
    A9(4) = 1
    A9(5) = 5
    A9(6) = 3
    A9(7) = 7
    GOTO 9420
   END IF
9411 IF NP = 2 THEN
    V9(6) = F(0)
    F(0) = F(1)
    F(1) = V9(6)    'Special case
   END IF
   ' ***** Set up structure table*****
   A9(0) = 0
   A9(1) = 1
   FOR I = 2 TO Q9
    FOR J = 0 TO (M9(I - 1) - 1)
        A9(J) = A9(J) + A9(J)
        A9(J + M9(I - 1)) = A9(J) + 1
    NEXT J
   NEXT I

9420 FOR I = 1 TO (C9(2) - 1)
    V9(4) = C9(2) * A9(I)
    V9(5) = I
    V9(6) = V9(4)
    V9(7) = F(V9(5))
    F(V9(5)) = F(V9(6))
    F(V9(6)) = V9(7)
    FOR J = 1 TO (A9(I) - 1)
        V9(5) = V9(5) + C9(2)
        V9(6) = V9(4) + A9(J)
        V9(7) = F(V9(5))
        F(V9(5)) = F(V9(6))
        F(V9(6)) = V9(7)
    NEXT J
   NEXT I
9500 '******    STAGES 1 & 2        ******
   '******  Get two-element DHTs   ******
   FOR I = 0 TO (NP - 2) STEP 2
    V9(6) = F(I) + F(I + 1)
    V9(7) = F(I) - F(I + 1)
    F(I) = V9(6)
    F(I + 1) = V9(7)
   NEXT I
  IF P = 1 THEN RETURN                    '
Finished

9510 '****** Get four-element DHTs *****
   FOR I = 0 TO (N - 4) STEP 4
    V9(6) = F(I) + F(I + 2)
    V9(7) = F(I + 1) + F(I + 3)
    V9(8) = F(I) - F(I + 2)
    V9(9) = F(I + 1) - F(I + 3)
    F(I) = V9(6)
    F(I + 1) = V9(7)
    F(I + 2) = V9(8)
    F(I + 3) = V9(9)
   NEXT I
9520 IF P = 2 THEN RETURN:                '
Finished
9600 '*******  STAGES 3 & 4 *********
   U9 = C9(6)
   S9 = 4
   FOR L9 = 2 TO C9(6)
    V9(2) = S9 + S9
    U9 = U9 - 1
```

```
      V9(3) = M9(U9 - 1)
      FOR Q9 = 0 TO C9(5) STEP V9(2)
          I = Q9
          D9 = I + S9
          V9(6) = F(I) + F(D9)
          V9(7) = F(I) - F(D9)
          F(I) = V9(6)
          F(D9) = V9(7)
          K9 = D9 - 1
          FOR J = V9(3) TO N9 STEP V9(3)
            I = I + 1
            D9 = I + S9
            E9 = K9 + S9
            V9(9) = F(D9) + F(E9) * T9(J)
            X9 = F(E9) - V9(9) * S9(J)
            Y9 = X9 * T9(J) + V9(9)
            V9(6) = F(I) + Y9
            V9(7) = F(I) - Y9
            V9(8) = F(K9) - X9
            V9(9) = F(K9) + X9
            F(I) = V9(6)
            F(D9) = V9(7)
            F(K9) = V9(8)
            F(E9) = V9(9)
            K9 = K9 - 1
          NEXT J
          E9 = K9 + S9
      NEXT Q9
      S9 = V9(2)
    NEXT L9
    N0 = NP
    DURA = TIMER - T0
    FOR I = 0 TO N - 1
        F(I) = F(I) / N
    NEXT I
    PRINT "TIEMPO = ", DURA
    PRINT "Presione cualquier tecla"
LB9: A$ = INKEY$: IF A$ = "" THEN GOTO LB9
    '//REORDENAR DATOS
    GOSUB READAF
RETURN

SPECTRUM:

'**************************************
'***************
    '          Subr GET POWER SPECTRUM

'**************************************
'***************
    GOSUB READAF
    D = 0
    T1 = TIMER
    P(0) = 2 * F(0) ^ 2
    NN = N / 2 - 1 - D
    FOR I = 1 TO NN
     P(I) = F(I) ^ 2 + F(N - I) ^ 2
    NEXT I
3000 ' Subr SMOOTH THE POWER SPECTRUM
    IF D = 0 THEN 4000
    FOR I = 1 TO D
        K = N / 2 - 1 - I
        FOR J = 0 TO K
         P(J) = P(J) + P(J + 1)
        NEXT J
        FOR J = 0 TO (K - 1)
         P(K - J) = P(K - J) + P(K - J - 1)
            NEXT J
            P(0) = 2 * P(0) ^ 2
        NEXT I
4000 :
    FOR I = 0 TO N / 2 - 1
     P(I) = (P(I) / 2) ^ .5
    NEXT I
    DURA1 = TIMER - T1
    CLS
    SCREEN 9
    DURAT = DURA + DURA1
    PRINT "TIEMPO = "; DURAT; " seg"
WT: ESPERA$ = INKEY$: IF ESPERA$ = "" THEN
GOTO WT
    RETURN

GSPECTRUM:
    CLS
    LOCATE 10, 20
    'INPUT "QUIERES EL ESPECTRO EN DECIBELES
(S/N) "; DEC$
    IF DEC$ = "S" AND P(0) > 0 THEN P(0) =
LOG(P(0))
    MAYOR = P(0)
    MENOR = P(0)
    FOR K = 1 TO NN
     IF DEC$ = "S" AND P(K) > 0 THEN P(K) =
LOG(P(K))
     P(K) = P(K)
     IF P(K) > MAYOR THEN MAYOR = P(K):
HMAY = K
     IF P(K) < MENOR THEN MENOR = P(K)
    NEXT K
    CLS
    SCREEN 9
    LOCATE 2, 1
    PRINT "                   VALOR MAYOR = ";
MAYOR
    PRINT "                   VALOR MENOR = ";
MENOR
'>>>>> PRINT "                  PERIODO = ";
N / HMAY
    LOCATE 20, 25
    PRINT "ESPECTRO DE POTENCIA PARA ";
NOMBRE$
    DY = MAYOR - MENOR
    '>>>>> IF DY = 0 THEN DY = MAYOR
    DY1 = (MAYOR * 320) / DY
    IF DY1 < 0 THEN DY1 = 0
    DX = 640 / NN
    LINE (1, 1)-(1, 400)
    LINE (1, DY1)-(640, DY1)
    FOR I = 0 TO NN
     y = (P(I) * 320) / DY
     Y1 = (P(I + 1) * 320) / DY
     y = (DY1 - y)
     Y1 = (DY1 - Y1)
     '>>>>> LINE (I * DX, Y)-((I + 1) * DX, Y1),
15:'ESPECTRO CONTINUO
     LINE (I * DX, DY1)-(I * DX, y), 15: 'ESPECTRO
DISCRETO
     LINE (I * DX, DY1 + 2)-(I * DX, DY1 - 2)
    NEXT I
    '//REORDENAR DATOS
    GOSUB READAF
LB7: A$ = INKEY$: IF A$ = "" THEN GOTO LB7
    RETURN
```

LISTA:

```
'*****************************************
***********
    'OBTENER VALORES MAXIMO Y MINIMO

'*****************************************
***********
    MAYOR = F(0)
    MENOR = F(0)
    FOR K = 1 TO N - 2
      IF F(K) > MAYOR THEN MAYOR = F(K)
      IF F(K) < MENOR THEN MENOR = F(K)
    NEXT K
    CLS
    FOR I = 0 TO N - 1
      PRINT "DATO "; I; " --> "; F(I)
      IF (I / 20) = INT(I / 20) AND I < > 0 THEN
            LOCATE 23, 10
            PRINT "<PRESIONA CUALQUIER
TECLA>"
ET1:   WTEC$ = INKEY$: IF WTEC$ = "" THEN
GOTO ET1
            CLS
      END IF

    NEXT I
    PRINT
    PRINT "           VALOR MAYOR = "; MAYOR
    PRINT "           VALOR MENOR = "; MENOR
    LOCATE 23, 10
    PRINT "<PRESIONA CUALQUIER TECLA>"
LB1: A$ = INKEY$: IF A$ = "" THEN GOTO LB1
    RETURN

E1:
    CLS
    COLOR 4, 0
    BEEP
    LOCATE 10, 20
    PRINT "ERROR <OPRIME RETURN>";
    INPUT AA$
    GOTO START
```

APENDICE 4

LISTADO DEL PROGRAMA FWT

```
DECLARE FUNCTION ESCRIBE! (A!)

'**************************************************************
    ' 05/09/91
    ' ANALISIS DE ESPECTRO POR
    ' TRANSFORMADA DE WALSH

'**************************************************************
    ' *DIMENSIONAR MATRICES *
    N1 = 8192
    DIM H(N1): ' //DATOS EN FUNCION DEL TIEMPO
    DIM F(N1): ' //DATOS EN FUNCION DE LA
FRECUENCIA
    DIM X(N1): ' //VECTOR DE PASO
    '// OPCIONES DEL MENU PRINCIPAL
    DIM OPCIONES$(10)
    OPCIONES$(1) = " 1.- CARGAR DATOS f(t)"
    OPCIONES$(2) = " 2.- CARGAR DATOS f(t) DE
ARCHIVO"
    OPCIONES$(3) = " 3.- CARGAR DATOS F(W) DE
ARCHIVO"
    OPCIONES$(4) = " 4.- SALVAR
TRANSFORMADA F(W)"
    OPCIONES$(5) = " 5.- OBTENER ESPECTRO"
    OPCIONES$(6) = " 6.- GRAFICAR DATOS f(t)
(TIEMPO)"
    OPCIONES$(7) = " 7.- GRAFICAR ESPECTRO"
    OPCIONES$(8) = " 8.- GENERAR FUNCION DE
WALSH"
    OPCIONES$(9) = " 9.- DESPLEGAR DATOS F(W)
"
    OPCIONES$(10) = "10.- SALIDA"
    PI = 3.141592653589795#

'************************************************************
    ' MENU DE OPCIONES

'************************************************************
START:
     CLS
     LOCATE 1, 18
     PRINT "ANALISIS DE ESPECTRO POR
TRANSFORMADA DE WALSH"
     LOCATE 3, 35
     PRINT "M E N U"
     FOR A = 1 TO 10
          LOCATE A + 5, 25
          COLOR 1, 0
          PRINT OPCIONES$(A)
     NEXT A
     LOCATE 6, 25
     COLOR 3, 0
     PRINT OPCIONES$(1)
     POSICION = 1
     SALIDA = 0

REP:
     TECLA$ = INKEY$
     IF TECLA$ = "" THEN GOTO REP
     IF ASC(MID$(TECLA$, 1, 1)) = 0 THEN
          SELECT CASE ASC(MID$(TECLA$, 2, 1))
           CASE 72
             COLOR 1, 0
             LOCATE POSICION + 5, 25
             PRINT OPCIONES$(POSICION)
             POSICION = POSICION - 1
             IF POSICION < = 0 THEN POSICION =
10
             COLOR 3, 0
             LOCATE POSICION + 5, 25
             PRINT OPCIONES$(POSICION)
           CASE 80
             COLOR 1, 0
             LOCATE POSICION + 5, 25
             PRINT OPCIONES$(POSICION)
             POSICION = POSICION + 1
             IF POSICION > 10 THEN POSICION = 1
             COLOR 3, 0
             LOCATE POSICION + 5, 25
             PRINT OPCIONES$(POSICION)
           CASE 71
             COLOR 1, 0
             LOCATE POSICION + 5, 25
             PRINT OPCIONES$(POSICION)
             POSICION = 1
             COLOR 3, 0
             LOCATE POSICION + 5, 25
             PRINT OPCIONES$(POSICION)
           CASE 79
             COLOR 1, 0
             LOCATE POSICION + 5, 25
             PRINT OPCIONES$(POSICION)
             POSICION = 10
             COLOR 3, 0
             LOCATE POSICION + 5, 25
             PRINT OPCIONES$(POSICION)
           CASE ELSE
          END SELECT
     END IF
     IF (ASC(TECLA$) = 13) THEN
          ON POSICION GOSUB PIDE, CARGA,
CARGA1, SALVA1, WALSH, GRAFICA, GRAFESP,
FUNCION, LISTA
          IF POSICION < > 10 THEN
            GOTO START
          ELSE
           SALIDA = 1
          END IF
     END IF
     IF (ASC(TECLA$) < > 27) AND (SALIDA = 0)
THEN GOTO REP
     END


PIDE:
```

```
'**************************************
'**************
  ' PEDIR DATOS
'**************************************
'**************
  CLS
  INPUT "Numero de puntos [1..8192]"; N
  P = INT(LOG(N) / LOG(2))
'**************************************
'*************************
  ' PEDIR DATOS DE LA TRASFORMADA
'**************************************
'*************************
  FOR I = 1 TO N
       PRINT "H("; I; ") ="; : INPUT H(I)
  NEXT I
  GOSUB SALVA: ' //SALVAR LOS DATOS
CAPTURADOS
  RETURN

SALVA:
'**************************************
'***********************
  'SALVAR LOS DATOS EN DOMINIO DEL TIEMPO
'**************************************
'***********************
  CLS
  LOCATE 1, 1
  INPUT "NOMBRE DEL ARCHIVO H(T) (*.T)";
NOMBRE$
  OPEN NOMBRE$ FOR OUTPUT AS #1
  PRINT #1, N
  FOR NPTS = 1 TO N
       A$ = STR$(H(NPTS))
       IF MID$(A$, 1, 1) = " " AND MID$(A$, 2,
1) = "." THEN
         A$ = "0" + MID$(A$, 2, LEN(A$) - 1)
       END IF
       IF MID$(A$, 1, 1) = "-" AND MID$(A$, 2,
1) = "." THEN
         A$ = "-0" + MID$(A$, 2, LEN(A$) - 1)
       END IF
       PRINT #1, A$
  NEXT NPTS
  CLOSE #1
  RETURN

SALVA1:
'**************************************
'***********************
  'SALVAR LOS DATOS EN DOMINIO DE LA
FRECUENCIA
'**************************************
'***********************
  CLS
  LOCATE 1, 1
  INPUT "NOMBRE DEL ARCHIVO H(W) (*.W)";
NOMBRE1$
  OPEN NOMBRE1$ FOR OUTPUT AS #1
  PRINT #1, N
  FOR NPTS = 1 TO N
       A$ = STR$(F(NPTS))
       IF MID$(A$, 1, 1) = " " AND MID$(A$, 2,
1) = "." THEN
         A$ = "0" + MID$(A$, 2, LEN(A$) - 1)
       END IF
       IF MID$(A$, 1, 1) = "-" AND MID$(A$, 2,
1) = "." THEN
         A$ = "-0" + MID$(A$, 2, LEN(A$) - 1)
       END IF
       PRINT #1, A$
  NEXT NPTS
  CLOSE #1
  RETURN

CARGA:
'**************************************
'************************
  ' CARGAR PUNTOS EN DOMINIO DEL TIEMPO
'**************************************
'************************
  CLS
  LOCATE 1, 1
  INPUT "NOMBRE DEL ARCHIVO H(T) [*.T]";
NOMBRE$
  OPEN NOMBRE$ FOR INPUT AS #1
  INPUT #1, NPT$
  N = VAL(NPT$)
  FOR NPTS = 1 TO N
       INPUT #1, CADENA$
       H(NPTS) = VAL(CADENA$)
  NEXT NPTS
  CLOSE #1
  ' // OBTENER ( POWER INDEX )
  P = INT(LOG(N) / LOG(2))
  RETURN

CARGA1:
'**************************************
'***********************
  'CARGAR DATOS EN DOMINIO DE LA
FRECUENCIA
'**************************************
'***********************
  CLS
  LOCATE 1, 1
  INPUT "NOMBRE DEL ARCHIVO H(F) [*.F]";
NOMBRE$
  OPEN NOMBRE$ FOR INPUT AS #1
  INPUT #1, NPT$
  N = VAL(NPT$)
  FOR NPTS = 1 TO N
  INPUT #1, CADENA$
       F(NPTS) = VAL(CADENA$)
  NEXT NPTS
  CLOSE #1
  ' // OBTENER POWER INDEX
```

```
    P = INT(LOG(N) / LOG(2))
    RETURN

GRAFICA:

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    'GRAFICAR LOS DATOS
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    CLS
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ' OBTENER EL MAXIMO Y EL MINIMO
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    MAYOR = H(1)
    MENOR = H(1)
    FOR K = 2 TO N - 1
    IF H(K) > MAYOR THEN MAYOR = H(K)
     IF H(K) < MENOR THEN MENOR = H(K)
    NEXT K
    '''''' AJUSTAR LIMITES PARA GRAFICAR
'''''''
    DY = MAYOR - MENOR
    IF DY = 0 THEN DY = MAYOR
    DY1 = (MAYOR * 320) / DY
    DX = 640 / N
    '''''' GRAFICAR ''''
    SCREEN 9
    ' //DESPLEGAR DATOS DE LA GRAFICA
    LOCATE 25, 5
    PRINT "F(t) = "; NOMBRE$; "   PUNTOS = "; N;
"        Y MAX = "; MAYOR; "   MIN = "; MENOR
    ' // DIBUJAR EL MARCO DE LA GRAFICA
    LINE (0, 0)-(639, 349), 14, B
    'LINE ((N + 2) / 2 * DX, 1)-((N + 2) / 2 * DX,
400), 14
    ' // DIBUJAR LINEA CON Y=0
    LINE (1, DY1 + 10)-(640, DY1 + 10), 14
    '''' GRAFICAR LOS PUNTOS ''
    FOR I = 1 TO N - 1
      Y = (H(I) * 320) / DY: Y1 = (H(I + 1) * 320) /
DY
      Y = (DY1 - Y)
      Y1 = (DY1 - Y1)
      LINE ((I - 1) * DX, Y + 10)-((I) * DX, Y1 + 10),
3
      LINE ((I - 1) * DX, DY1 + 13)-((I - 1) * DX, DY1
+ 7), 4
    NEXT I
LB2: A$ = INKEY$: IF A$ = "" THEN GOTO LB2
    RETURN

GRAFESP:

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ' REDONDEO DE LOS DATOS
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    'GOSUB readaf
    CLS
    PRINT "NUMERO DE PUNTOS = "; N
    INPUT "LIMITE MINIMO "; XMIN
    INPUT "LIMITE MAXIMO "; XMAX
    IF XMAX = 0 THEN
      XMIN = 1
      XMAX = N
    END IF
'''''''''''''''''''''''''''''''''''''''''''''''''
    'OBTENER VALORES MAXIMO Y MINIMO
'''''''''''''''''''''''''''''''''''''''''''''''''
    MAYOR = F(XMIN)
    MENOR = F(XMAX)
    FOR K = XMIN + 1 TO XMAX - 1
     IF F(K) > MAYOR THEN MAYOR = F(K)
     IF F(K) < MENOR THEN MENOR = F(K)
    NEXT K
    CLS
    SCREEN 9
    LOCATE 25, 5
    PRINT "TRANSF DE F(t) = "; NOMBRE$; "   N ="; 
N
    LOCATE 25, 41
    PRINT "Y MAX = "; MAYOR; "  MIN = "; MENOR
'''''''''''''''''''''''''''''''''''''''''''''''''
    ' OBTENER LAS RELACIONES NECESARIAS PARA
GRAFICAR
'''''''''''''''''''''''''''''''''''''''''''''''''
    DY = MAYOR - MENOR
    IF DY = 0 THEN DY = MAYOR
    DY1 = (MAYOR * 320) / DY
    DX = 640 / (XMAX + 1 - XMIN)
    '// DIBUJAR MARCO DE LA GRAFICA
    LINE (0, 0)-(639, 349), 14, B
    LINE ((N / 2 - XMIN) * DX, 1)-((N / 2 - XMIN) * DX,
400), 14
    '// DIBUJAR LINEA Y=0
    LINE (1, DY1 + 10)-(640, DY1 + 10), 14
    '// GRAFICAR LOS PUNTOS
    FOR I = XMIN TO XMAX
      Y = (F(I) * 320) / DY: Y1 = (F(I + 1) * 320) /
DY
      Y = (DY1 - Y): Y1 = (DY1 - Y1)

      ''''LINEA OPCIONAL ''''
      'LINE ((I - XMIN) * DX, Y+10)-(((I - XMIN) + 1)
* DX, Y1+10), 3:'ESPECTRO CONTINUO
      ''''''''''''''''''''''''

      LINE ((I - XMIN) * DX, DY1 + 10)-((I - XMIN) *
DX, Y + 10), 3: 'ESPECTRO DISCRETO
      LINE ((I - XMIN) * DX, DY1 + 13)-((I - XMIN) *
DX, DY1 + 7), 4
    NEXT I
    'GOSUB readaf
LB4: A$ = INKEY$: IF A$ = "" THEN GOTO LB4
    RETURN
```

```
WALSH:
*********************************************
*******
      'SUBRUTINA PARA CALCULAR LA
TRANSFORMADA RAPIDA
      'DE WALSH

*********************************************
*******

  CLS
  PRINT "CALCULANDO"
  TO = TIMER
  FOR I = 1 TO N
    F(I) = H(I)
  NEXT I
  N2 = N / 2
  P = LOG(N) / LOG(2)
  FOR L = 1 TO P
   NY = 0
   NZ = 2 ^ (L - 1)
   NZI = 2 * NZ
   NZN = N / NZI
   FOR I = 1 TO NZN
    NX = NY + 1
    NY = NY + NZ
    JS = (I - 1) * NZI
    JD = JS + NZI + 1
    FOR J = NX TO NY
         JS = JS + 1
         J2 = J + N2
         X(JS) = F(J) + F(J2)
         JD = JD - 1
         X(JD) = F(J) - F(J2)
    NEXT J
   NEXT I
   FOR K = 1 TO N
        F(K) = X(K)
   NEXT K
  NEXT L
  FOR K = 1 TO N
   F(K) = F(K) / N
  NEXT K
  T1 = TIMER - TO
  PRINT "TIEMPO DE CALCULO = "; T1
LB5: A$ = INKEY$: IF A$ = "" THEN GOTO LB5
RETURN


FUNCION:

*********************************************
***
      'SUBRUTINA PARA CALCULAR FUNCIONES DE
WALSH

*********************************************
***

      CLS
      PRINT "NUMERO DE FUNCION DE WALSH"; :
INPUT W
      PRINT "NUMERO DE PUNTOS"; : INPUT N
      PRINT "NOMBRE DE LA FUNCION"; : INPUT
NOMBRE$
      E = W
      NN = LOG(W) / LOG(2)
      FOR I = 0 TO NN
          B(I) = E - INT(E / 2) * 2
          E = INT(E / 2)
      NEXT I
      FOR J = 0 TO N - 1
      MM = 1
           FOR I = 1 TO NN
            MM = MM * COS(2 ^ I * PI * J / (N - 1)) ^
B(I)
           NEXT I
      H(J + 1) = SGN((SIN(2 * PI * J / (N - 1)) ^ B(0)
* MM))
      NEXT J
      H(1) = H(2)
      H(J) = H(J - 1)
      PRINT "QUIERES SALVAR LA FUNCION QUE
GENERASTE (1/SI,ENTER/NO)"; : INPUT SALVAR
      IF SALVAR = 1 THEN GOSUB SALVA
RETURN

LISTA:

*********************************************
************
    'OBTENER VALORES MAXIMO Y MINIMO

*********************************************
************
    MAYOR = F(1)
    MENOR = F(1)
    FOR K = 2 TO N - 2
     IF F(K) > MAYOR THEN MAYOR = F(K)
     IF F(K) < MENOR THEN MENOR = F(K)
    NEXT K
    CLS
    FOR I = 1 TO N
     PRINT "DATO "; I - 1; " = "; F(I)
     IF (I / 20) = INT(I / 20) AND I <> 0 THEN
          LOCATE 23, 10
          PRINT "<PRESIONA CUALQUIER TECLA>"
ET1:    WTEC$ = INKEY$: IF WTEC$ = "" THEN
GOTO ET1
          CLS
     END IF
    NEXT I
    PRINT
    PRINT "   QUIERES VER ALGUN VALOR
(1/SI,ENTER/NO)": INPUT VALOR
    IF VALOR > 0 THEN
         PRINT "NUMERO DEL DATO = ": INPUT I
         PRINT "F("; I; ")="; F(I + 1)
    END IF
    PRINT
    PRINT "           VALOR MAYOR = "; MAYOR
    PRINT "           VALOR MENOR = "; MENOR
    LOCATE 23, 10
    PRINT "<PRESIONA CUALQUIER TECLA>"
LB9: A$ = INKEY$: IF A$ = "" THEN GOTO LB9
    RETURN

FUNCTION ESCRIBE (A)


END FUNCTION
```

APENDICE 5

LISTADO DEL PROGRAMA FHAT

```
DECLARE FUNCTION ESCRIBE! (A!)
'**************************************************************
    ' 05/09/92
    ' ANALISIS DE ESPECTRO POR
    ' TRANSFORMADA DE HAAR

'**************************************************************
    ' '*DIMENSIONAR MATRICES '
    N1 = 4096
    DIM H(N1): ' //DATOS EN FUNCION DEL TIEMPO
    DIM F(N1): ' //DATOS EN FUNCION DE LA
FRECUENCIA
    DIM I(N1)
    '// OPCIONES DEL MENU PRINCIPAL
    DIM OPCIONES$(10)
    OPCIONES$(1) = " 1.- CARGAR DATOS f(t)"
    OPCIONES$(2) = " 2.- CARGAR DATOS f(t) DE
ARCHIVO"
    OPCIONES$(3) = " 3.- CARGAR DATOS HA(f)
DE ARCHIVO"
    OPCIONES$(4) = " 4.- SALVAR
TRANSFORMADA HA(F)"
    OPCIONES$(5) = " 5.- OBTENER ESPECTRO"
    OPCIONES$(6) = " 6.- GRAFICAR DATOS f(t)
(TIEMPO)"
    OPCIONES$(7) = " 7.- GRAFICAR ESPECTRO"
    OPCIONES$(8) = " 8.- GENERAR FUNCION DE
HAAR"
    OPCIONES$(9) = " 9.- DESPLEGAR DATOS
HA(f) "
    OPCIONES$(10) = "10.- SALIDA"
    PI = 3.141592653589795#

'**************************************************************
    ' MENU DE OPCIONES

'**************************************************************
START:
    CLS
    LOCATE 1, 18
    PRINT "ANALISIS DE ESPECTRO POR
TRANSFORMADA DE HAAR"
    LOCATE 3, 35
    PRINT "OPCIONES"
    FOR A = 1 TO 10
        LOCATE A + 5, 25
        COLOR 1, 0
        PRINT OPCIONES$(A)
    NEXT A
    LOCATE 6, 25
    COLOR 3, 0
    PRINT OPCIONES$(1)
    POSICION = 1
    SALIDA = 0
REP:
    TECLA$ = INKEY$
    IF TECLA$ = "" THEN GOTO REP
    IF ASC(MID$(TECLA$, 1, 1)) = 0 THEN
        SELECT CASE ASC(MID$(TECLA$, 2, 1))
         CASE 72
          COLOR 1, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
          POSICION = POSICION - 1
         IF POSICION <= 0 THEN POSICION = 10
          COLOR 3, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
         CASE 80
          COLOR 1, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
          POSICION = POSICION + 1
          IF POSICION > 10 THEN POSICION = 1
          COLOR 3, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
         CASE 71
          COLOR 1, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
          POSICION = 1
          COLOR 3, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
         CASE 79
          COLOR 1, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
          POSICION = 10
          COLOR 3, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
         CASE ELSE
        END SELECT
    END IF
    IF (ASC(TECLA$) = 13) THEN
        ON POSICION GOSUB PIDE, CARGA,
CARGA1, SALVA1, HAAR, GRAFICA, GRAFESP,
FUNCION, LISTA
        IF POSICION <> 10 THEN
         GOTO START
        ELSE
         SALIDA = 1
        END IF
    END IF
    IF (ASC(TECLA$) <> 27) AND (SALIDA = 0)
THEN GOTO REP
    END

PIDE:
'**************************************************************
    ' PEDIR DATOS
```

```
'****************************************
'************************
    CLS
    INPUT "Numero de puntos [1..8192]"; N
    P = INT(LOG(N) / LOG(2))

'****************************************
'************************
    ' PEDIR DATOS DE LA TRASFORMADA

'****************************************
'************************
    FOR I = 1 TO N
     PRINT "H("; I; ") ="; : INPUT H(I)
    NEXT I
    PRINT "QUIERES SALVAR LOS DATOS
CAPTURADOS (1/SI,ENTER/NO)"; : INPUT SALVAR
    IF SALVAR = 1 THEN GOSUB SALVA: '
//SALVAR LOS DATOS CAPTURADOS
    RETURN


SALVA:

'****************************************
'************************
    'SALVAR LOS DATOS EN DOMINIO DEL TIEMPO

'****************************************
'************************
    CLS
    LOCATE 1, 1
    INPUT "NOMBRE DEL ARCHIVO H(T) (*.T)";
NOMBRE$
    OPEN NOMBRE$ FOR OUTPUT AS #1
    PRINT #1, N
    FOR NPTS = 1 TO N
        A$ = STR$(H(NPTS))
        IF MID$(A$, 1, 1) = " " AND MID$(A$, 2,
1) = "." THEN
          A$ = "0" + MID$(A$, 2, LEN(A$) - 1)
        END IF
        IF MID$(A$, 1, 1) = "-" AND MID$(A$, 2,
1) = "." THEN
          A$ = "-0" + MID$(A$, 2, LEN(A$) - 1)
        END IF
        PRINT #1, A$
    NEXT NPTS
    CLOSE #1
    RETURN


SALVA1:

'****************************************
'************************
    'SALVAR LOS DATOS EN DOMINIO DE LA
FRECUENCIA

'****************************************
'************************
    CLS
    LOCATE 1, 1
    INPUT "NOMBRE DEL ARCHIVO HA(F) (*.HA)";
NOMBRE1$
    OPEN NOMBRE1$ FOR OUTPUT AS #1
    PRINT #1, N
    FOR NPTS = 1 TO N
        A$ = STR$(F(NPTS))
        IF MID$(A$, 1, 1) = " " AND MID$(A$, 2,
1) = "." THEN
          A$ = "0" + MID$(A$, 2, LEN(A$) - 1)
        END IF
        IF MID$(A$, 1, 1) = "-" AND MID$(A$, 2,
1) = "." THEN
          A$ = "-0" + MID$(A$, 2, LEN(A$) - 1)
        END IF
        PRINT #1, A$
    NEXT NPTS
    CLOSE #1
    RETURN

CARGA:
'****************************************
'************************
    ' CARGAR PUNTOS EN DOMINIO DEL TIEMPO

'****************************************
'************************
    CLS
    LOCATE 1, 1
    INPUT "NOMBRE DEL ARCHIVO H(T) [*.T]";
NOMBRE$
    OPEN NOMBRE$ FOR INPUT AS #1
    INPUT #1, NPT$
    N = VAL(NPT$)
    FOR NPTS = 1 TO N
        INPUT #1, CADENA$
        H(NPTS) = VAL(CADENA$)
    NEXT NPTS
    CLOSE #1
    ' // OBTENER ( POWER INDEX )
    P = INT(LOG(N) / LOG(2))
    RETURN

CARGA1:
'****************************************
'************************
    'CARGAR DATOS EN DOMINIO DE LA
FRECUENCIA

'****************************************
'************************
    CLS
    LOCATE 1, 1
    INPUT "NOMBRE DEL ARCHIVO HA(F) [*.HA]";
NOMBRE$
    OPEN NOMBRE$ FOR INPUT AS #1
    INPUT #1, NPT$
    N = VAL(NPT$)
    FOR NPTS = 1 TO N
        INPUT #1, CADENA$
        F(NPTS) = VAL(CADENA$)
    NEXT NPTS
    CLOSE #1
    ' // OBTENER POWER INDEX
    P = INT(LOG(N) / LOG(2))
    RETURN

GRAFICA:

'****************************************
'************************
```

```
'GRAFICAR LOS DATOS
'**************************************
****************************
CLS
'**************************************
****************************
' OBTENER EL MAXIMO Y EL MINIMO
'**************************************
****************************
MAYOR = H(1)
MENOR = H(1)
FOR K = 2 TO N - 1
 IF H(K) > MAYOR THEN MAYOR = H(K)
 IF H(K) < MENOR THEN MENOR = H(K)
NEXT K
'***** AJUSTAR LIMITES PARA GRAFICAR
*******
DY = MAYOR - MENOR
IF DY = 0 THEN DY = MAYOR
DY1 = (MAYOR * 320) / DY
DX = 630 / N
'***** GRAFICAR ****
SCREEN 9
' //DESPLEGAR DATOS DE LA GRAFICA
LOCATE 25, 5
PRINT "F(t) = "; NOMBRE$; "   PUNTOS = "; N;
"     Y MAX = "; MAYOR; "   MIN = "; MENOR
' // DIBUJAR EL MARCO DE LA GRAFICA
LINE (0, 0)-(639, 349), 14, B
LINE ((N + 2) / 2 * DX, 1)-((N + 2) / 2 * DX,
400), 14
' // DIBUJAR LINEA CON Y=0
LINE (1, DY1 + 10)-(640, DY1 + 10), 14
'*** GRAFICAR LOS PUNTOS **
FOR I = 1 TO N - 1
  Y = (H(I) * 320) / DY: Y1 = (H(I + 1) * 320) /
DY
  Y = (DY1 - Y)
  Y1 = (DY1 - Y1)
  LINE (I * DX, Y + 10)-((I + 1) * DX, Y1 + 10),
3
  LINE (I * DX, DY1 + 13)-(I * DX, DY1 + 7), 4
NEXT I
LB2: A$ = INKEY$: IF A$ = "" THEN GOTO LB2
RETURN

GRAFESP:
'**************************************
*************
' SUBRUTINA PARA GRAFICAR EL ESPECTRO
'**************************************
*************
' GOSUB READAF
CLS
PRINT "NUMERO DE PUNTOS = "; N
INPUT "LIMITE MINIMO "; XMIN
INPUT "LIMITE MAXIMO "; XMAX
IF XMAX = 0 THEN
  XMIN = 1
  XMAX = N
END IF
'**************************************
************
'OBTENER VALORES MAXIMO Y MINIMO
'**************************************
************
MAYOR = F(XMIN)
MENOR = F(XMAX)
FOR K = XMIN + 1 TO XMAX - 1
  IF F(K) > MAYOR THEN MAYOR = F(K)
  IF F(K) < MENOR THEN MENOR = F(K)
NEXT K
CLS
SCREEN 9
LOCATE 25, 5
PRINT "TRANSF DE F(t) = "; NOMBRE$; "   N
="; N
LOCATE 25, 41
PRINT "Y MAX = "; MAYOR; "  MIN = ";
MENOR
'**************************************
***********
' OBTENER LAS RELACIONES NECESARIAS PARA
GRAFICAR
'**************************************
************
DY = MAYOR - MENOR
IF DY = 0 THEN DY = MAYOR
DY1 = (MAYOR * 320) / DY
DX = 635 / (XMAX + 1 - XMIN)
'// DIBUJAR MARCO DE LA GRAFICA
LINE (0, 0)-(639, 349), 14, B
LINE ((NP / 2 - XMIN) * DX, 1)-((NP / 2 - XMIN) *
DX, 400), 14
'// DIBUJAR LINEA Y=0
LINE (1, DY1 + 10)-(640, DY1 + 10), 14
'// GRAFICAR LOS PUNTOS
FOR I = XMIN TO XMAX - 1
  Y = (F(I) * 320) / DY: Y1 = (F(I + 1) * 320) /
DY
  Y = (DY1 - Y): Y1 = (DY1 - Y1)

  '***LINEA OPCIONAL ****
  'LINE ((I - XMIN) * DX, Y+10)-(((I - XMIN) + 1)
* DX, Y1+10), 3:'ESPECTRO CONTINUO
  '*********************

  LINE ((I - XMIN) * DX, DY1 + 10)-((I - XMIN) *
DX, Y + 10), 3: 'ESPECTRO DISCRETO
  LINE ((I - XMIN) * DX, DY1 + 13)-((I - XMIN) *
DX, DY1 + 7), 4
NEXT I
' GOSUB READAF
LB4: A$ = INKEY$: IF A$ = "" THEN GOTO LB4
RETURN

HAAR:
'**************************************
*********************
' SUBRUTINA PARA CALCULAR LA
TRANSFORMADA DE HAAR
'**************************************
*********************
```

```
  FOR I = 1 TO N
   F(I) = H(I)
  NEXT I
  CLS
  PRINT "CALCULANDO"
  TO = TIMER
  P = LOG(N) / LOG(2)
  FOR I = 1 TO P
   L = P + 1 - I
   L2 = 2 ^ (L - 1)
   FOR Z = 1 TO 2 * L2
         I(Z) = F(Z)
   NEXT Z
   FOR J = 1 TO L2
         L3 = L2 + J
         JJ = 2 * J - 1
         F(J) = I(JJ) + I(JJ + 1)
         F(L3) = I(JJ) - I(JJ + 1)
   NEXT J
  NEXT I
  FOR I = 1 TO N
         IF I > 1 THEN PP = INT(LOG(I - 1) /
LOG(2))
         F(I) = F(I) / N * ((2 ^ .5) ^ PP)
  NEXT I
  T1 = TIMER - TO
  BEEP
  PRINT "Tiempo : "; T1
LB5: A$ = INKEY$: IF A$ = "" THEN GOTO LB5
RETURN

FUNCION:
'*************************************
'*********************
         '       SUBRUTINA PARA GENERAR
FUNCIONES DE HAAR

'*************************************
'*********************
         CLS
         PRINT : PRINT "NUMERO DE LA FUNCION
DE HAAR A GENERAR "; : INPUT W
         PRINT "NUMERO DE PUNTOS "; : INPUT N
         PRINT "NOMBRE DEL ARCHIVO"; : INPUT
NOMBRE$
         FOR I = 1 TO N
          H(I) = 0
         NEXT I
         LW = INT(LOG(W) / LOG(2))
         R = W - 2 ^ LW
         L1 = N * R / 2 ^ LW
         L2 = N * (R + 1 / 2) / 2 ^ LW
         FOR L = L1 TO L2 - 1
          H(L + 1) = (2 ^ .5) ^ LW
         NEXT L
         L3 = N * (R + 1) / 2 ^ LW
         FOR L = L2 TO L3 - 1
          H(L + 1) = -(2 ^ .5) ^ LW
         NEXT L
         PRINT "QUIERES SALVAR EL ARCHIVO
(1/SI,ENTER/NO)"; : INPUT SALVAR
         IF SALVAR = 1 THEN GOSUB SALVA
RETURN

LISTA:
'*************************************
'***********
    'OBTENER VALORES MAXIMO Y MINIMO

'*************************************
'***********
    MAYOR = F(1)
    MENOR = F(1)
    FOR K = 2 TO N - 2
     IF F(K) > MAYOR THEN MAYOR = F(K)
     IF F(K) < MENOR THEN MENOR = F(K)
    NEXT K
    CLS
    FOR I = 1 TO N
     PRINT "DATO "; I - 1; " = "; F(I)
     IF (I / 20) = INT(I / 20) AND I <> 0 THEN
          LOCATE 23, 10
          PRINT "<PRESIONA CUALQUIER
TECLA>"
ET1:    WTEC$ = INKEY$: IF WTEC$ = "" THEN
GOTO ET1
          CLS
     END IF
    NEXT I
    PRINT
    PRINT "     QUIERES VER ALGUN VALOR EN
ESPECIAL (ENTER/NO,1/SI)": INPUT VALOR
    IF VALOR > 0 THEN
     INPUT "NUMERO DEL DATO = ", I
     PRINT "DATO "; I; " = "; F(I + 1)
    END IF
    PRINT
    PRINT "          VALOR MAYOR = "; MAYOR
    PRINT "          VALOR MENOR = "; MENOR
    LOCATE 23, 10
    PRINT "<PRESIONA CUALQUIER TECLA>"
LB9: A$ = INKEY$: IF A$ = "" THEN GOTO LB9
    RETURN

FUNCTION ESCRIBE (A)


END FUNCTION
```

APENDICE 6

LISTADO DEL PROGRAMA MEM

```
DECLARE FUNCTION ESCRIBE! (A!)
'***************************************
' 24/05/92
    ' ANALISIS DE ESPECTRO POR
    ' METODO DE MAXIMA ENTROPIA
'***************************************
' 'DIMENSIONAR MATRICES '
    N1 = 4096
    DIM H(N1): ' //DATOS EN FUNCION DEL TIEMPO
    DIM F(N1): ' //DATOS EN FUNCION DE LA
FRECUENCIA
    DIM I(N1)
    DIM FASE1(N1)
    DIM COSE(N1)
    DIM SENO(N1)
    DIM BUFERE(N1)
    DIM COF(N1)
    DIM WK1(N1)
    DIM WK2(N1)
    DIM WKM(N1)
    '// OPCIONES DEL MENU PRINCIPAL
    DIM OPCIONES$(10)
    OPCIONES$(1) = " 1.- CARGAR DATOS f(t)"
    OPCIONES$(2) = " 2.- CARGAR DATOS f(t) DE
ARCHIVO"
    OPCIONES$(3) = " 3.- CARGAR DATOS F(f) DE
ARCHIVO"
    OPCIONES$(4) = " 4.- SALVAR
TRANSFORMADA H(F)"
    OPCIONES$(5) = " 5.- OBTENER ESPECTRO"
    OPCIONES$(6) = " 6.- GRAFICAR DATOS f(t)
(TIEMPO)"
    OPCIONES$(7) = " 7.- GRAFICAR ESPECTRO"
    OPCIONES$(8) = " 8.- GRAFICAR FASE"
    OPCIONES$(9) = " 9.- DESPLEGAR DATOS F(F)"
    OPCIONES$(10) = "10.- SALIDA"
    PI = 3.141592653589795#
'***************************************
    ' MENU DE OPCIONES
'***************************************
START:
    CLS
    LOCATE 1, 22
    PRINT "ANALISIS DE ESPECTRO POR MAXIMA
ENTROPIA"
    LOCATE 3, 35
    PRINT "OPCIONES"
    FOR A = 1 TO 10
        LOCATE A + 5, 25
        COLOR 1, 0
        PRINT OPCIONES$(A)
    NEXT A
    LOCATE 6, 25
    COLOR 3, 0
    PRINT OPCIONES$(1)
    POSICION = 1
    SALIDA = 0
REP:
    TECLA$ = INKEY$
IF TECLA$ = "" THEN GOTO REP
    IF ASC(MID$(TECLA$, 1, 1)) = 0 THEN
        SELECT CASE ASC(MID$(TECLA$, 2, 1))
         CASE 72
          COLOR 1, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
          POSICION = POSICION - 1
         IF POSICION < = 0 THEN POSICION = 10
          COLOR 3, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
         CASE 80
          COLOR 1, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
          POSICION = POSICION + 1
          IF POSICION > 10 THEN POSICION = 1
          COLOR 3, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
         CASE 71
          COLOR 1, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
          POSICION = 1
          COLOR 3, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
         CASE 79
          COLOR 1, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
          POSICION = 10
          COLOR 3, 0
          LOCATE POSICION + 5, 25
          PRINT OPCIONES$(POSICION)
         CASE ELSE
        END SELECT
    END IF
    IF (ASC(TECLA$) = 13) THEN
        ON POSICION GOSUB PIDE, CARGA,
CARGA1, SALVA1, 9000, GRAFICA, GRAFESP,
FASE, LISTA
        IF POSICION < > 10 THEN
         GOTO START
        ELSE
         SALIDA = 1
        END IF
    END IF
    IF (ASC(TECLA$) < > 27) AND (SALIDA = 0)
THEN GOTO REP
    END


PIDE:
'***************************************
******************
```

```
   ' PEDIR DATOS
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
   CLS
   INPUT "Numero de puntos [1..8192]"; N
   P = INT(LOG(N) / LOG(2))
'''''''''''''''''''''''''''''''''''''''
   ' PEDIR DATOS DE LA TRASFORMADA
'''''''''''''''''''''''''''''''''''''''
   FOR I = 1 TO N
    PRINT "H("; I; ") ="; : INPUT H(I): ' COS(18 *
PI * I / n) + 5 * RND
   NEXT I
   GOSUB SALVA: ' //SALVAR LOS DATOS
CAPTURADOS
   RETURN

SALVA:
''''''''''''''''''''''''''''''''''''
   'SALVAR LOS DATOS EN DOMINIO DEL TIEMPO
''''''''''''''''''''''''''''''''''''''''''''''''''''
CLS
   LOCATE 1, 1
   INPUT "NOMBRE DEL ARCHIVO H(T) (*.T)";
NOMBRE$
   OPEN NOMBRE$ FOR OUTPUT AS #1
   PRINT #1, N
   FOR NPTS = 1 TO N
        A$ = STR$(H(NPTS))
        IF MID$(A$, 1, 1) = " " AND MID$(A$, 2,
1) = "." THEN A$ = "0" + MID$(A$, 2, LEN(A$)-1)
        END IF
        IF MID$(A$, 1, 1) = "-" AND MID$(A$, 2,
1) = "." THEN A$ = "-0" +MID$(A$, 2, LEN(A$)-1)
        END IF
        PRINT #1, A$
   NEXT NPTS
   CLOSE #1
   RETURN

SALVA1:
''''''''''''''''''''''''''''''''''''
'SALVAR LOS DATOS EN DOMINIO DE LA
FRECUENCIA
''''''''''''''''''''''''''''''''''''
   CLS
   LOCATE 1, 1
   INPUT "NOMBRE DEL ARCHIVO H(F) (*.M)";
NOMBRE1$
   OPEN NOMBRE1$ FOR OUTPUT AS #1
   PRINT #1, N
   FOR NPTS = 1 TO N
        A$ = STR$(F(NPTS))
        IF MID$(A$, 1, 1) = " " AND MID$(A$, 2,
1) = "." THEN
          A$ = "0" + MID$(A$, 2, LEN(A$) - 1)
        END IF
        IF MID$(A$, 1, 1) = "-" AND MID$(A$, 2,
1) = "." THEN
          A$ = "-0" + MID$(A$, 2, LEN(A$) - 1)
        END IF
        PRINT #1, A$
   NEXT NPTS
   CLOSE #1
   RETURN

CARGA:
''''''''''''''''''''''''''''''''''''''''''''''''''''
' CARGAR PUNTOS EN DOMINIO DEL TIEMPO
''''''''''''''''''''''''''''''''''''''''''''''''''''
CLS
   LOCATE 1, 1
   INPUT "NOMBRE DEL ARCHIVO H(T) (*.T)";
NOMBRE$
   OPEN NOMBRE$ FOR INPUT AS #1
   INPUT #1, NPT$
   N = VAL(NPT$)
   FOR NPTS = 1 TO N
        INPUT #1, CADENA$
        H(NPTS) = VAL(CADENA$)
   NEXT NPTS
   CLOSE #1
   ' // OBTENER ( POWER INDEX )
   P = INT(LOG(N) / LOG(2))
   RETURN

CARGA1:
''''''''''''''''''''''''''''''''''''''''''''''''''''
'CARGAR DATOS EN DOMINIO DE LA FRECUENCIA
''''''''''''''''''''''''''''''''''''''''''''''''''''
CLS
   LOCATE 1, 1
   INPUT "NOMBRE DEL ARCHIVO H(F) (*.M)";
NOMBRE$
   OPEN NOMBRE$ FOR INPUT AS #1
   INPUT #1, NPT$
   N = VAL(NPT$)
   FOR NPTS = 1 TO N
        INPUT #1, CADENA$
        F(NPTS) = VAL(CADENA$)
   NEXT NPTS
   CLOSE #1
   ' // OBTENER POWER INDEX
   P = INT(LOG(N) / LOG(2))
   RETURN

GRAFICA:
''''''''''''''''''''''''''''''''''''''''''''''''''''
'GRAFICAR LOS DATOS
''''''''''''''''''''''''''''''''''''''''''''''''''''
CLS
''''''''''''''''''''''''''''''''''''''''''''''''''''
' OBTENER EL MAXIMO Y EL MINIMO
''''''''''''''''''''''''''''''''''''''''''''''''''''
   MAYOR = H(1)
   MENOR = H(1)
   FOR K = 2 TO N - 1
    IF H(K) > MAYOR THEN MAYOR = H(K)
    IF H(K) < MENOR THEN MENOR = H(K)
   NEXT K
   '***** AJUSTAR LIMITES PARA GRAFICAR
   DY = MAYOR - MENOR
   IF DY = 0 THEN DY = MAYOR
   DY1 = (MAYOR * 320) / DY
   DX = 620 / N
   '***** GRAFICAR ****
   SCREEN 9
   ' //DESPLEGAR DATOS DE LA GRAFICA
   LOCATE 25, 5
   PRINT "F(t) = "; NOMBRE$; "   PUNTOS = "; N;
"    Y MAX = "; MAYOR; "   MIN = "; MENOR
   ' // DIBUJAR EL MARCO DE LA GRAFICA
```

```
   LINE (0, 0)-(639, 349), 14, B
   LINE (N / 2 * DX, 1)-(N / 2 * DX, 400), 14
   ' // DIBUJAR LINEA CON Y=0
   LINE (1, DY1 + 10)-(640, DY1 + 10), 14
   '*** GRAFICAR LOS PUNTOS **
   FOR I = 1 TO N - 1
  Y = (H(I) * 320) / DY: Y1 = (H(I + 1) * 320) / DY
    Y = (DY1 - Y)
    Y1 = (DY1 - Y1)
  LINE (I * DX, Y + 10)-((I + 1) * DX, Y1 + 10), 3
  LINE (I * DX, DY1 + 13)-(I * DX, DY1 + 7), 4
   NEXT I
LB2: A$ = INKEY$: IF A$ = "" THEN GOTO LB2
   RETURN

reada:
'************************************
'   REACOMODAR DATOS
'************************************
FOR I = 1 TO N / 2
     T = H(I + N / 2)
     H(I + N / 2) = H(I)
     H(I) = T
   NEXT I
   RETURN

readaf:
'************************************
' REORDENAR DATOS PARA GRAFICAR
'************************************
FOR I = 1 TO N / 2
    T = F(I + N / 2)
    F(I + N / 2) = F(I)
    F(I) = T
  NEXT I
  RETURN

GRAFESP:
'************************************
' REDONDEO DE LOS DATOS
'************************************
GOSUB readaf
  CLS
  PRINT "NUMERO DE PUNTOS = "; NP
  INPUT "LIMITE MINIMO "; xmin
  INPUT "LIMITE MAXIMO "; xmax
  IF xmax = 0 THEN
   xmin = 1
   xmax = N
  END IF
  'FOR I = 0 TO (N - 1)
  '>>>>>>>>>>> F(I) = INT(.5 + 1000 / N *
F(I)) / 1000
  'NEXT I

'**************************************
'***********
  'OBTENER VALORES MAXIMO Y MINIMO

'**************************************
'***********
  MAYOR = F(xmin)
  MENOR = F(xmax)
  FOR K = xmin + 1 TO xmax - 1
   IF F(K) > MAYOR THEN MAYOR = F(K)
   IF F(K) < MENOR THEN MENOR = F(K)
  NEXT K
   CLS
   SCREEN 9
   LOCATE 25, 5
   PRINT "TRANSF DE F(t) = "; NOMBRE$; "   N
="; NP
   LOCATE 25, 41
   PRINT "Y MAX = "; MAYOR; " MIN = ";
MENOR

'**************************************
'***********
   ' OBTENER LAS RELACIONES NECESARIAS PARA
GRAFICAR

'**************************************
'***********
   DY = MAYOR - MENOR
   IF DY = 0 THEN DY = MAYOR
   DY1 = (MAYOR * 320) / DY
   DX = 635 / (xmax + 1 - xmin)
   '// DIBUJAR MARCO DE LA GRAFICA
   LINE (0, 0)-(639, 349), 14, B
   LINE ((NP / 2 - xmin) * DX, 1)-((NP / 2 - xmin) *
DX, 400), 14
   '// DIBUJAR LINEA Y=0
   LINE (1, DY1 + 10)-(640, DY1 + 10), 14
   '// GRAFICAR LOS PUNTOS
   FOR I = xmin TO xmax - 1
     Y = (F(I) * 320) / DY: Y1 = (F(I + 1) * 320) /
DY
     Y = (DY1 - Y): Y1 = (DY1 - Y1)

     '***LINEA OPCIONAL ****
     'LINE ((I - XMIN) * DX, Y+10)-(((I - XMIN) + 1)
* DX, Y1+10), 3:'ESPECTRO CONTINUO
    '*********************
     LINE ((I - xmin) * DX, DY1 + 10)-((I - xmin) *
DX, Y + 10), 3: 'ESPECTRO DISCRETO
     LINE ((I - xmin) * DX, DY1 + 13)-((I - xmin) *
DX, DY1 + 7), 4
    NEXT I
    GOSUB readaf
LB4: A$ = INKEY$: IF A$ = "" THEN GOTO LB4
   RETURN

9000 : 'const
  CLS
  PRINT : PRINT : PRINT
  INPUT "Numero de polos"; MP
  INPUT "Incremento en frecuencia"; INC
  INPUT "Frecuencia final"; FF
  IF INC = 0 THEN INC = 1 / N
  IF FF = 0 THEN FF = 1
  TIEMPO = TIMER
  P = 0
  FOR J = 1 TO N
   P = P + H(J) ^ 2
  NEXT J
  PM = P / N
  WK1(1) = H(1)
  WK2(N - 1) = H(N)
  FOR J = 2 TO N - 1
   WK1(J) = H(J)
   WK2(J - 1) = H(J)
  NEXT J
FOR K = 1 TO MP
   NUM = 0
```

```
    DENOM = 0
    FOR J = 1 TO N - K
     NUM = NUM + WK1(J) * WK2(J)
    DENOM = DENOM + WK1(J) ^ 2 + (WK2(J)) ^2
    NEXT J
    COF(K) = 2 * NUM / DENOM
    PM = PM * (1 - (COF(K)) ^ 2)
    FOR I = 1 TO K - 1
     COF(I) = WKM(I) - COF(K) * WKM(K - I)
    NEXT I
    IF K = M THEN GOTO MEM
    FOR I = 1 TO K
     WKM(I) = COF(I)
    NEXT I
    FOR J = 1 TO N - K - 1
     WK1(J) = WK1(J) - WKM(K) * WK2(J)
    WK2(J) = WK2(J + 1) - WKM(K) * WK1(J + 1)
    NEXT J
NEXT K
MEM:
        NP = FF / INC
        FOR IT = 1 TO NP
                fdt = INC * FF * IT
                theta = 2 * PI * fdt
                wpr = COS(theta)
                wpi = SIN(theta)
                wr = 1
                wi = 0
                sumr = 1
                sumi = 0
                FOR I = 1 TO MP
                wtemp = wr
                wr = wr * wpr - wi * wpi
                wi = wi * wpr + wtemp * wpi
                wrs = wr
                wis = wi
                sumr = sumr - COF(I) * wrs
                sumi = sumi - COF(I) * wis
          EVLMEM = PM / ((sumr) ^ 2 + (sumi) ^ 2)
                NEXT I
                F(IT) = EVLMEM
        NEXT IT
        TIEMPO = TIMER - TIEMPO
        BEEP
        PRINT "Tiempo: "; TIEMPO
LB5: A$ = INKEY$: IF A$ = "" THEN GOTO LB5
RETURN
FASE:
     IF FASE(K) > MAYOR THEN MAYOR =
FASE(K)
     IF FASE(K) < MENOR THEN MENOR = FASE(K)
    'NEXT k
    LOCATE 25, 5
    PRINT "TRANSF DE F(t) = "; NOMBRE$; "   N";N
    LOCATE 25, 41
    PRINT "Y MAX = "; MAYOR; "  MIN = ";
MENOR
''''''''''''''''''''''''''''''''''''''''
    ' OBTENER LAS RELACIONES NECESARIAS
PARA GRAFICAR
''''''''''''''''''''''''''''''''''''''''
    DY = MAYOR - MENOR: DY1 = (MAYOR * 320)
/ DY: DX = 635 / (xmax + 1 - xmin)
    LINE (0, 0)-(639, 349), 14, B
    LINE ((N / 2 - xmin) * DX, 1)-((N / 2 - xmin) * DX,
400), 14
    LINE (1, DY1 + 10)-(640, DY1 + 10), 14
    FOR I = xmin TO xmax - 1
     Y = (FASE(I) * 320) / DY: Y1 = (FASE(I + 1) *
320) / DY
     Y = (DY1 - Y): Y1 = (DY1 - Y1)
     LINE ((I - xmin) * DX, Y + 10)-(((I - xmin) + 1)
* DX, Y1 + 10), 3
     LINE ((I - xmin) * DX, DY1 + 13)-((I - xmin) *
DX, DY1 + 7), 4
    NEXT I
LB7: A$ = INKEY$: IF A$ = "" THEN GOTO LB7
    RETURN
LISTA:
''''''''''''''''''''''''''''''''''''''''
    'OBTENER VALORES MAXIMO Y MINIMO

''''''''''''''''''''''''''''''''''''''''
    MAYOR = F(1)
    MENOR = F(1)
    FOR K = 2 TO N - 2
     IF F(K) > MAYOR THEN MAYOR = F(K)
     IF F(K) < MENOR THEN MENOR = F(K)
    NEXT K
    CLS
    FOR I = 1 TO N
     PRINT "DATO "; I; " = "; F(I)
     IF (I / 20) = INT(I / 20) AND I <> 0 THEN
         LOCATE 23, 10
         PRINT "<PRESIONA CUALQUIER
TECLA>"
ET1:    WTEC$ = INKEY$: IF WTEC$ = "" THEN
GOTO ET1
         CLS
     END IF
    NEXT I
    PRINT
    PRINT "          VALOR MAYOR = "; MAYOR
    PRINT "          VALOR MENOR = "; MENOR
    PRINT
    PRINT "          TIEMPO UTILIZADO = ";
TIEMPO
    LOCATE 23, 10
    PRINT "<PRESIONA CUALQUIER TECLA>"
LB9: A$ = INKEY$: IF A$ = "" THEN GOTO LB9
    RETURN

FUNCTION ESCRIBE (A)

END FUNCTION
```