

## 5. PRINCIPIOS EN PROCESAMIENTO DE SEÑALES

En el presente apartado se expondrá el procesamiento estándar que se puede realizar a una señal adquirida.

### 5.1. ¿Por qué procesar?

El procesamiento de señales es importante porque permite saber qué propiedades tiene un conjunto de datos adquiridos, en sí, permite llevar una señal a una forma en la cual se pueda interpretar mejor.

Una herramienta importante en el procesamiento de datos es el filtrado. El filtrado se realiza para adecuar una señal recibida para ser interpretada de una mejor forma o limpiarla para poderla apreciar más claramente. Los filtros ayudan en este sentido, eliminando componentes de frecuencia indeseadas o que no interesan.

Entre otras herramientas de procesamiento conocidas se encuentra el análisis de Fourier, el cual permite cambiar el dominio en el que se encuentran los datos. Hacer un análisis de Fourier sin herramientas computacionales es una labor compleja y desgastante, sobretodo cuando se necesita alta precisión en la respuesta y cuando la cantidad de datos usados en el cómputo es elevada. Continuamente se han creado algoritmos que permiten simplificar el cálculo de la Transformada de Fourier, cada uno de ellos con ventajas en lo que se refiere al tiempo de procesamiento y simplicidad de los cálculos.

Se expondrá el camino que se recorrió desde la transformada de Fourier de tiempo discreto hasta el algoritmo de Transformada Rápida de Fourier (FFT, del inglés *Fast Fourier Transform*)

### 5.2. Los filtros y su importancia para un procesamiento correcto

En algunas aplicaciones resulta necesario reducir o eliminar ciertas amplitudes de frecuencia de las señales. Este proceso se conoce como filtrado.

Según su efecto, los filtros se dividen en dos categorías:

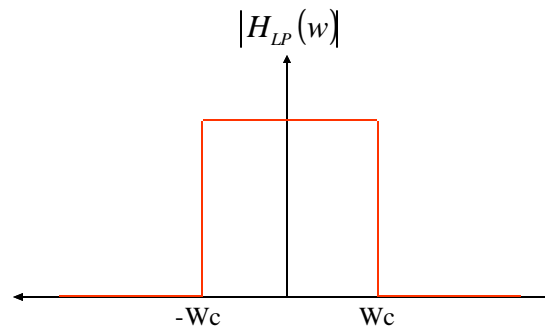
- Conformatores de frecuencia: Los que cambian el espectro
- Selectivos en frecuencia: los que eliminan ciertas frecuencias y dejan pasar otras.

Según su implementación, los filtros se dividen en:

- Análogos:
  - Pasivos: No necesitan alimentación
    - Se hacen con resistencias, condensadores y bobinas
    - Elevada impedancia de entrada, con lo cual se carga la etapa anterior y la ganancia depende de la frecuencia
    - Impedancia de salida baja, la salida carga el filtro
    - Uso de inductancias elevadas para filtrar en baja frecuencia.
  - Activos
    - Necesitan alimentación
    - Impedancia de entrada grande
    - Impedancia de salida muy baja
    - Resistencias y condensadores como elementos constitutivos
    - El margen dinámico de voltaje de salida está limitado por la saturación de los amplificadores operacionales
    - El ancho de banda es limitado
    - Los amplificadores operacionales generan ruido
- Digitales
  - *FIR: Finite Impulse Response*: Son muy estables porque no tienen polos, además tienen fase lineal bajo ciertas condiciones. Pueden diseñarse filtros con pendientes muy pronunciadas, haciéndolo casi un filtro ideal.
  - *IIR: Infinite Impulse Response*: Es el tipo de filtro digital resultante de aproximar un filtro análogo.

Según su función, los filtros se clasifican en:

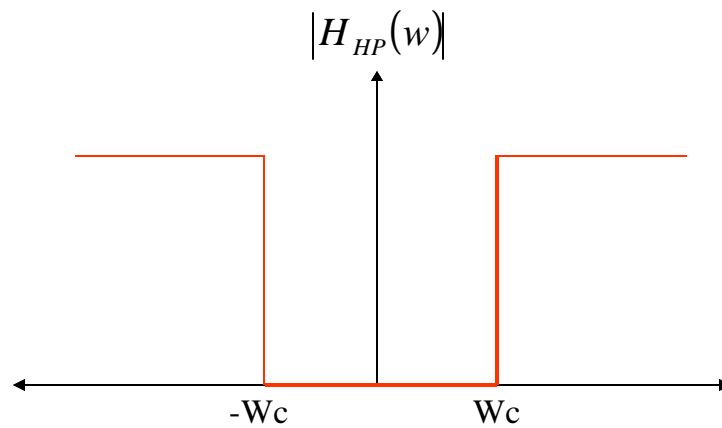
- Pasabajo



$$|H_{LP}(w)| = \begin{cases} 1, & |w| < Wc \\ 0, & |w| > Wc \end{cases}$$

**Figura 35.** Función de ganancia de un filtro Pasabajos

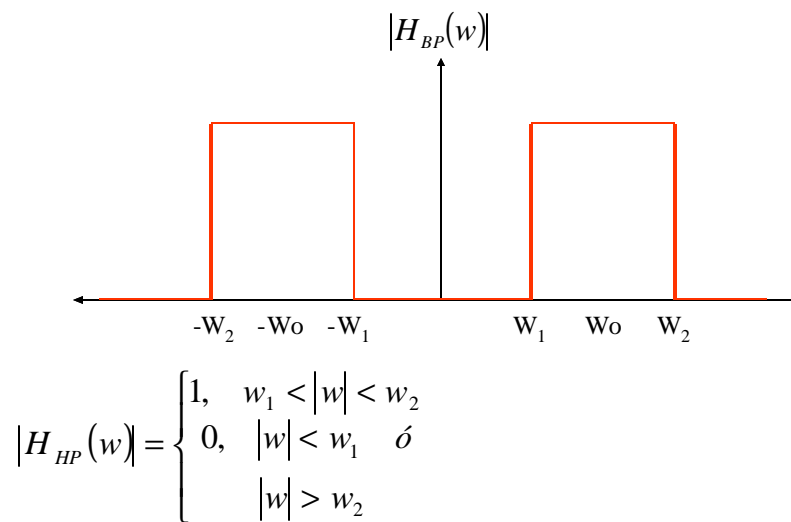
- Pasaalto:



$$|H_{HP}(w)| = \begin{cases} 0, & |w| < Wc \\ 1, & |w| > Wc \end{cases}$$

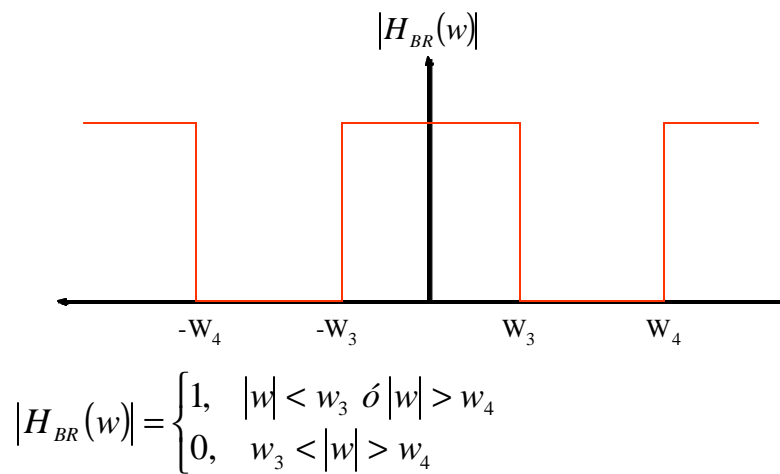
**Figura 36.** Función de ganancia de un filtro Pasaaltos

- Filtro Pasabanda



**Figura 37.** Función de ganancia de un filtro Pasabanda

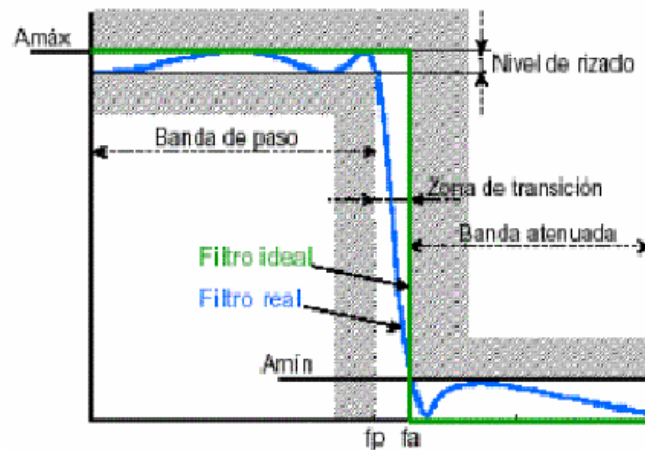
- Filtro Eliminabanda



**Figura 38.** Función de ganancia de un filtro Eliminabanda

Los filtros son muy útiles a la hora de solucionar problemas relacionados con el tratamiento de señales.

Entre los parámetros más importantes que tiene un filtro se encuentran los relacionados en las figuras 39 y 40:



**Figura 39.** Parámetros importantes de un filtro en el dominio de la frecuencia.

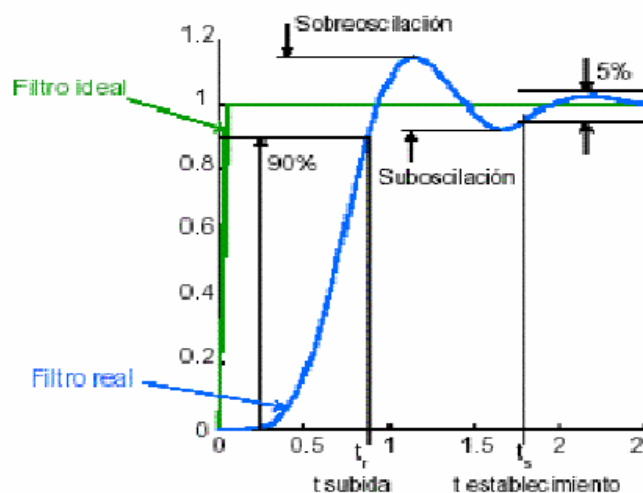
FUENTE: DOMINIO DE LA FRECUENCIA. En: Aplicación de los filtros Analógicos. [En línea] España, 2000

<<http://bellota.ele.uva.es/~lourdes/docencia/Sensores/Trabajos0304/FiltrosAnalogicos.pdf>>

[Consulta: 18 de Enero de 2004]

En el dominio de la frecuencia:

- Ancho de la zona de transición
- Atenuación en la banda de paso
- Atenuación en banda de atenuación
- Nivel de rizado
- Desfase de frecuencia



**Figura 40.** Parámetros importantes de un filtro en el dominio del tiempo

FUENTE: DOMINIO DEL TIEMPO. En: Aplicación de los filtros Analógicos. [En línea] España, 2000

<<http://bellota.ele.uva.es/~lourdes/docencia/Sensores/Trabajos0304/FiltrosAnalogicos.pdf>>

[Consulta: 18 de Enero de 2004]

En el dominio del tiempo:

- Tiempo de subida (90%)
- Tiempo de establecimiento (5%)
- Sobreoscilación
- Suboscilación

Además de la aproximación que se use para implementar un filtro, es recomendable hacerlo con un orden de filtro elevado. Un filtro ideal necesitaría un orden infinito, pero es imposible de fabricar.

La función de transferencia del filtro sólo es afectada por el tiempo de aproximación a usar.

Las aproximaciones de filtros más usadas son:

- Respuesta de *Butterworth*: Siendo sus características más importantes:
  - La ganancia es lo más plana posible en la banda de paso a expensas de un corte poco abrupto.
  - Pendiente de transición no demasiado abrupta
  - Respuesta transitoria aceptable.
- Respuesta de *Chebyshev*
  - Corte más abrupto a expensas de rizado en la banda de paso.
  - Pendiente de transición más alta (cerca de la frecuencia de paso ( $f_p$ ))
  - Respuesta transitoria peor que *Butterworth*.
- Respuesta de *Bessel*
  - Mínima deformación de la señal, presenta una respuesta que se aproxima a la ideal
  - Desfase lineal con la frecuencia. Tiempo de propagación constante.
  - Pendiente de transición peor que *Butterworth* y *Chebyshev* para un mismo orden
- Respuesta elíptica
  - Corte muy abrupto con rizado en la banda de paso y en la atenuada.
  - Respuesta transitoria peor que *Chebyshev*.
  - Adecuados para la eliminación de una frecuencia concreta.

### 5.3. La transformada Discreta de Fourier:

La transformada Discreta de Fourier (DFT, del inglés *Discrete Fourier Transform*) se deriva del análisis de Fourier en tiempo continuo. Esta demostración está fuera de los propósitos de este trabajo, así que se aceptará la formula que se describe a continuación:

en donde:

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

$$W_N = e^{-j(2\pi/N)}$$

En la ecuación (1),  $X(k)$  es el vector de datos en la frecuencia y  $x(n)$  es el vector de datos en el tiempo, con la posibilidad de ser números complejos. Los tiempos de computación de esta expresión para hallar la Transformada Discreta de Fourier son muy altos, debido a las operaciones matemáticas que se deben realizar. Se han desarrollado algoritmos que permiten un tiempo de cómputo más bajo sacrificando poca precisión en el resultado. Entre los algoritmos más importantes se encuentran:

- Algoritmo de Goertzel
- Decimación en el tiempo
- Decimación en frecuencia
- FFT radix-2
- FFT radix-4
- Algoritmo de transformada de Hartley

A continuación se detallará el algoritmo de *Decimación en el tiempo* y luego el de la *Transformada de Hartley*.

#### 5.3.1. Decimación en el tiempo:

La eficiencia de este algoritmo radica en la descomposición de la DFT en DFT's más pequeñas, de menor número de términos. Este proceso se puede hacer gracias a la simetría y la periodicidad de la exponencial compleja vista anteriormente:

$$W_N^{kn} = e^{-j(2\pi/N)kn}$$

Como primer paso, la secuencia original de datos muestreados en el tiempo  $x(n)$  se debe dividir en subsecuencias más pequeñas. Para ilustrar ésto, se considera un número total de muestras  $N$ . El número de muestras  $N$  debe ser potencia de 2, así:  $N=2^v$ . Como se sabe que  $N$  es un número par, se puede separar la secuencia original  $x(n)$  en dos secuencias cada una de  $N/2$  elementos. La mejor forma es ubicar en una secuencia las componentes pares y en otra las impares. Esto se ve mejor matemáticamente así:

$$X(k) = \sum_{n \text{ par}} x(n) W_N^{nk} + \sum_{n \text{ impar}} x(n) W_N^{nk} \quad \text{Ecuación (2)}$$

Sustituyendo las variables  $n=2r$  si  $n$  es par y  $n=2r+1$  para  $n$  impar en la ecuación anterior, se obtiene:

$$\begin{aligned} X(k) &= \sum_{r=0}^{(N/2)-1} x(2r) W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x(2r+1) W_N^{(2r+1)k} \\ X(k) &= \sum_{r=0}^{(N/2)-1} x(2r) (W_N^2)^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x(2r+1) (W_N^2)^{rk} \quad \text{Ecuación (3)} \end{aligned}$$

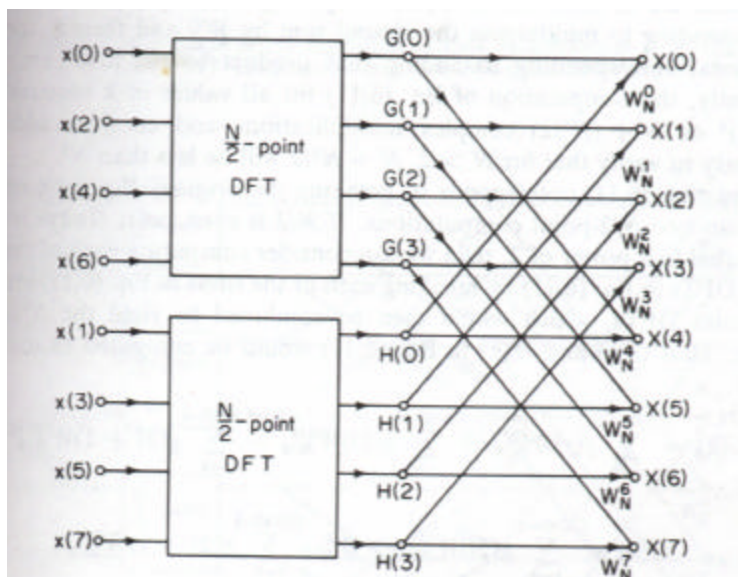
Pero como  $W_N^2 = W_{N/2}$ , se puede expresar la ecuación anterior de la siguiente forma:

$$\begin{aligned} X(k) &= \sum_{r=0}^{(N/2)-1} x(2r) W_{N/2}^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x(2r+1) W_{N/2}^{rk} \\ X(k) &= G(k) + W_N^k H(k) \quad \text{Ecuación (4)} \end{aligned}$$

Cada una de las sumas en la ecuación (4) corresponde a una DFT; la primera sumatoria de los números pares y la segunda de los impares. A pesar que el rango de  $K$  va desde 0 hasta  $N-1$ , cada una de las sumas se computa solamente para  $k$  entre 0 y  $(N/2)-1$ .

Luego que la dos DTF se han desarrollado, éstas se combinan para obtener la DTF de  $N$  puntos. La figura 41 muestra cómo se conforma:





**Figura 41.** Diagrama de flujo que muestra el algoritmo de decimación en el tiempo para obtener la DFT de  $N$  puntos partiendo de su división en dos DFT de  $N/2$  puntos.

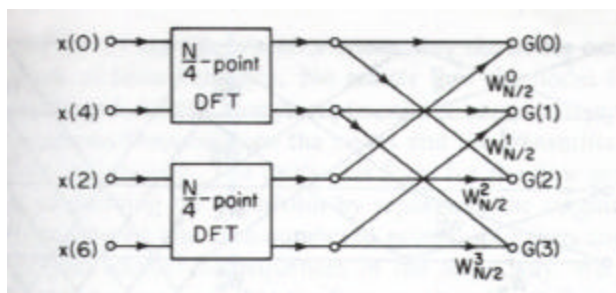
FUENTE: OPPENHEIM, Alan. Digital Signal Processing. Estados Unidos. Prentice-Hall, 1975.

585 paginas

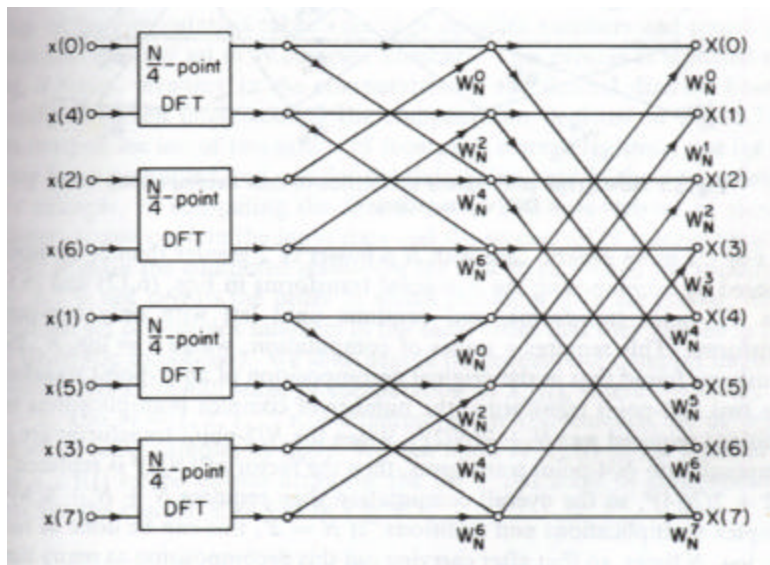
[Consulta: 4 de agosto de 2004]

En el diagrama se observa que  $G(k)$  representa la DFT de 4 puntos de los números pares y  $H(k)$  representa la DFT de 4 puntos de los números impares (esto se infiere de la ecuación (4)).

El siguiente paso sería dividir la DFT de 4 puntos en 2 DFT de dos puntos. La figura 42 muestra este procedimiento:



a.



b.

**Figura 42.** a) Decimación en el tiempo para una DTF de  $N/2$  puntos en dos DFT de  $N/4$  puntos.

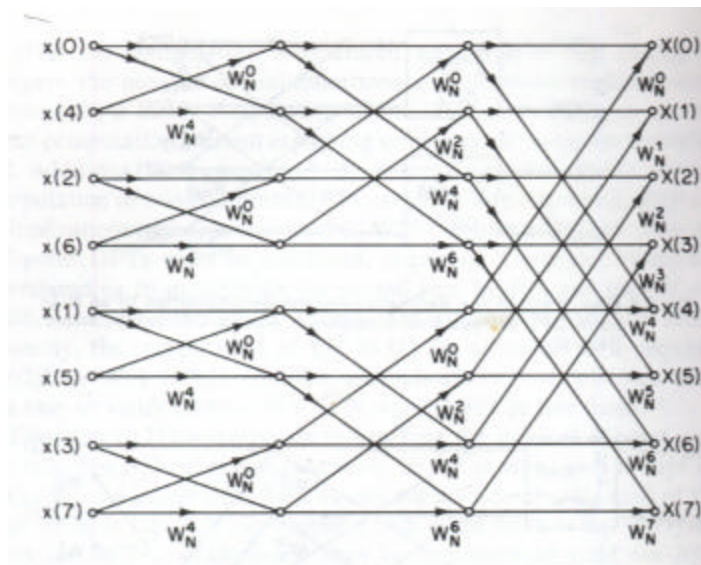
b) La gráfica del apartado (a) se adjunta a la vista anteriormente en la figura 41.

FUENTE: OPPENHEIM, Alan. Digital Signal Processing. Estados Unidos. Prentice-Hall, 1975.

585 paginas

[Consulta: 4 de agosto de 2004]

Por medio del algoritmo de decimación en el tiempo, una DFT de 8 puntos se convierte en una DTF de 2 puntos, como se muestra en la figura 43:



**Figura 43.** Gráfico completo de la decimación en el tiempo de una DFT de 8 puntos.

FUENTE: OPPENHEIM, Alan. Digital Signal Processing. Estados Unidos. Prentice-Hall, 1975.

585 paginas [Consulta: 4 de agosto de 2004]

En un acaso más general con  $N$  puntos, se puede proceder a descomponer la DFT de  $N/4$  puntos en  $N/8$  puntos, y continuar haciendo ese tipo de divisiones hasta dejar tan solo DFT de 2 puntos. Esto requiere  $v$  etapas de cómputo, donde  $v = \log_2 N$

La decimación en el tiempo es un procedimiento útil porque de una transformada inicial muy compleja, difícil de resolver en un solo paso, pasa a resolver múltiples transformadas más sencillas de 2 términos.

5.3.1.1. Ordenamiento del vector de datos inicial en el tiempo: Mirando más detalladamente la tabla 1, se observa que los datos en la parte izquierda del diagrama se encuentran desordenados, esto es como consecuencia de la división de cada secuencia en sus términos pares e impares. Aparentemente está desordenada, pero en verdad se encuentra ordenada en lo que se denomina *orden de bit inverso*. Esto se refiere a coger cada elemento y trasladar los bits más significativos a la parte menos significativa, y viceversa. Para explicar mejor ésto, se tiene el siguiente ejemplo para  $N=8$ :

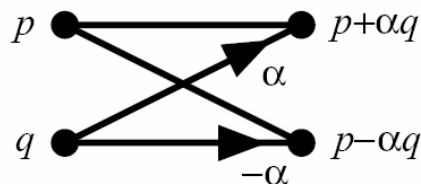
Orden entrada	Argumento	Codificación binaria	Codificación Binaria Inv.	Nuevo argumento	Orden salida
F(0)	0	000	000	0	F(0)
F(4)	4	100	001	1	F(1)
F(2)	2	010	010	2	F(2)
F(6)	6	110	011	3	F(3)
F(1)	1	001	100	4	F(4)
F(5)	5	101	101	5	F(5)
F(3)	3	011	110	6	F(6)
F(7)	7	111	111	7	F(7)

**Tabla 1.** Ordenamiento de los datos en orden de bit inverso

El reordenamiento en orden de bit inverso de la secuencia  $X(n)$  es necesario por la manera como el cómputo de la DFT se descompone en DFT sucesivas más pequeñas.

5.3.1.2. Cómputos tipo mariposa:

El corazón del algoritmo para hallar la DFT de una secuencia es la computación de tipo mariposa. Se llama así por que su forma gráfica describe la forma de este insecto. La figura 44 describe este procedimiento:



**Figura 44.** Cómputo tipo mariposa.

FUENTE: CARNEGIE MELLON UNIVERSITY. Butterfly calculation. [En Línea] Estados Unidos, 2004 <<http://www-2.cs.cmu.edu/afs/andrew/scs/cs/15-463/pub/www/notes/fourier/fourier.pdf>>

[Consulta: agosto 20 de 2004]

En este gráfico la letra griega  $\alpha$  representa al término  $W_N$ .

### 5.3.2. Transformada de Hartley:

La expresión para hallar la transformada de Hartley es muy parecida a la que se usa para transformada de Fourier, excepto por la unidad imaginaria  $i$ , la cual no se encuentra. Ésta transformada empieza con una secuencia de muestras en el tiempo. Se tiene la secuencia  $X(t)$  con transformada de Hartley  $H(f)$ :

$$H(f) = \frac{1}{N} \sum_{t=0}^{N-1} X(t) \left[ \cos\left(\frac{2p f t}{N}\right) + \sin\left(\frac{2p f t}{N}\right) \right] \quad \text{Ecuación (5)}$$

Mientras que la transformada de Fourier en su definición tiene una parte compleja, la transformada de Hartley sólo contempla en su definición números reales. Esto es indiferente para el mundo real, en donde los datos que se recolectan son reales. La finalidad de la Transformada de Hartley en este trabajo de grado es disminuir la complejidad computacional y los tiempos de procesamiento.

Al igual que con la Transformada Rápida de Fourier, se procede a dividir la secuencia de datos real en secuencias más pequeñas para reducir la complejidad de las iteraciones y los cálculos. Se dividen en secuencias par e impar:

$$X_0(t) = X(2t)$$

$$X_1(t) = X(2t + 1)$$

Se considerarán los índices como módulo N. Pero desde que  $X_0(t)$  y  $X_1(t)$  sean definidos en términos de  $2t$ , repiten módulo  $N/2$ ; por tanto, si  $X_0(t)$  y  $X_1(t)$  son secuencias de longitud  $N/2$  se obtiene entonces:

$$\begin{aligned} H_0(f) &= \frac{1}{M} \sum_{t=0}^{N-1} X_0(t) \left[ \cos\left(\frac{2\pi f t}{M}\right) + \sin\left(\frac{2\pi f t}{M}\right) \right] \\ H_1(f) &= \frac{1}{M} \sum_{t=0}^{N-1} X_1(t) \left[ \cos\left(\frac{2\pi f t}{M}\right) + \sin\left(\frac{2\pi f t}{M}\right) \right] \end{aligned}, \text{ con } M = N/2 \quad \text{Ecuación (6)}$$

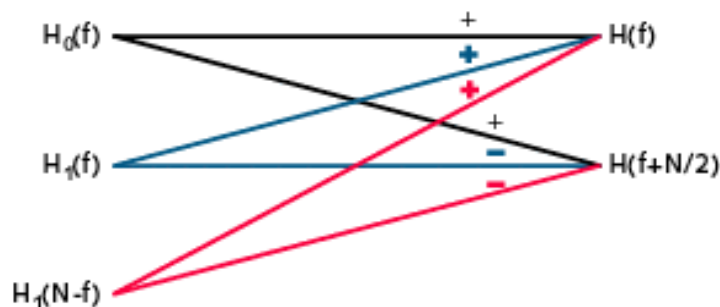
$X_0(t)$  se refiere a los términos pares y  $X_1(t)$  se refiere a los términos impares de la secuencia  $X(t)$ . Operando algebraicamente estas dos ecuaciones anteriores con las identidades trigonométricas apropiadas se obtiene:

$$H(f) = \frac{1}{2} \left\{ H_0(f) + H_1(f) \cos\left(\frac{2\pi f}{N}\right) + H_1(N-f) \sin\left(\frac{2\pi f}{N}\right) \right\} \quad \text{Ecuación (7)}$$

Y por simetría:

$$H\left(f + \frac{N}{2}\right) = \frac{1}{2} \left\{ H_0(f) - H_1(f) \cos\left(\frac{2\pi f}{N}\right) - H_1(N-f) \sin\left(\frac{2\pi f}{N}\right) \right\} \quad \text{Ecuación (8)}$$

Estas dos ecuaciones definitivas (7) y (8) definen la computación mariposa vista anteriormente, como se muestra en la siguiente figura 45:



**Figura 45.** Una mariposa simple de la transformada de Hartley.

FUENTE: SCOTT, Robert. En: Embedded Systems Programming Magazine [En Línea] Estados Unidos, 2003

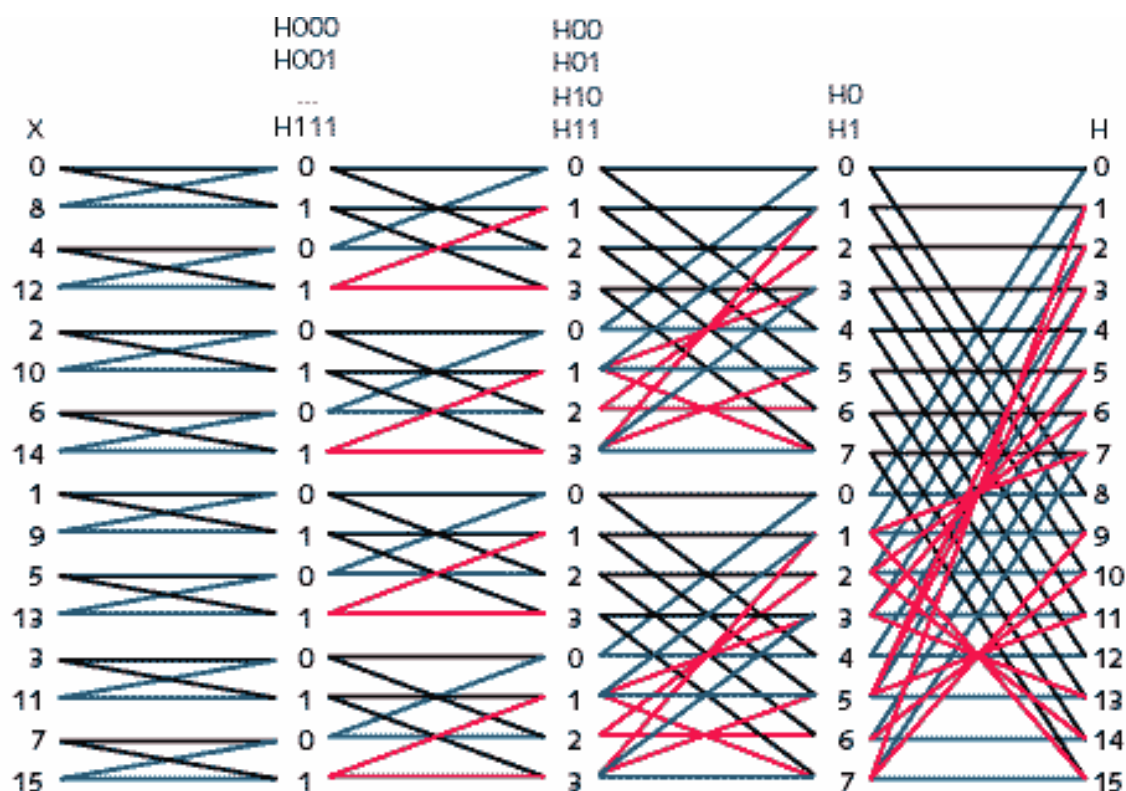
<<http://www.embedded.com/2000/0009/0009feat3.htm>>

[Consulta: abril 20 de 2004]

Como convención para interpretar correctamente la figura 45, los cálculos con líneas de color azul representan términos que tienen un factor coseno. Las líneas rojas representan términos que cuentan con un factor seno. La línea negra representa términos que no cuentan con factores trigonométricos.

Este diagrama ilustra la transformada de Hartley de  $N$  términos en función de dos transformadas de longitud  $N/2$ . El proceso continúa hasta que queden sólo transformadas de Hartley de longitud 1.

Para apreciar todo este procedimiento de una mejor forma, se cuenta con la figura 46. Esta figura representa un ejemplo de una transformada de Hartley con  $N=16$ . Dado a que es este el método empleado para realizar el análisis en frecuencia de la señal ECG, se explicará de forma detallada a continuación.



**Figura 46.** Una mariposa simple de la transformada de Hartley

FUENTE: SCOTT, Robert. En: Embedded Systems Programming Magazine [En Línea] Estados Unidos, 2003

<<http://www.embedded.com/2000/0009/0009feat3.htm>>

[Consulta: abril 20 de 2004]

La figura 46 muestra el diagrama de flujo que ilustra la descomposición recursiva de una transformada de Hartley de 16 datos. A pesar que el flujo de la información va de izquierda a derecha, se analizará de derecha a izquierda

En el lado derecho, se observa una columna de números desde el cero al quince. Éstos números representan los valores desde  $H(0)$  hasta  $H(15)$ , los cuales simbolizan la transformada de Hartley de la secuencia original en el dominio del tiempo desde  $X(0)$  hasta  $X(15)$ .

Si en las dos últimas ecuaciones (7) y (8) halladas se dan valores a  $f$  desde 0 hasta 7, se observa que valores desde  $H(0)$  hasta  $H(15)$  se pueden obtener a partir de dos transformadas de Hartley obtenidas previamente, de longitud  $N=8$ , de la siguiente forma:

$$H(0) = \frac{1}{2} \{H_0(0) + H_1(0)\cos(0) + H_1(0)\sen(0)\}$$

$$H(8) = \frac{1}{2} \{H_0(0) - H_1(0)\cos(0) - H_1(0)\sen(0)\}$$

si  $f=0$

$$H(1) = \frac{1}{2} \left\{ H_0(1) + H_1(1)\cos\left(\frac{p}{8}\right) + H_1(7)\sen\left(\frac{p}{8}\right) \right\}$$

$$H(9) = \frac{1}{2} \left\{ H_0(1) - H_1(1)\cos\left(\frac{p}{8}\right) - H_1(7)\sen\left(\frac{p}{8}\right) \right\}$$

si  $f=1$

$$H(2) = \frac{1}{2} \left\{ H_0(2) + H_1(2)\cos\left(\frac{p}{4}\right) + H_1(6)\sen\left(\frac{p}{4}\right) \right\}$$

$$H(10) = \frac{1}{2} \left\{ H_0(2) - H_1(2)\cos\left(\frac{p}{4}\right) - H_1(6)\sen\left(\frac{p}{4}\right) \right\}$$

si  $f=2$

$$H(3) = \frac{1}{2} \left\{ H_0(3) + H_1(3)\cos\left(\frac{3p}{8}\right) + H_1(5)\sen\left(\frac{3p}{8}\right) \right\}$$

$$H(11) = \frac{1}{2} \left\{ H_0(3) - H_1(3)\cos\left(\frac{3p}{8}\right) - H_1(5)\sen\left(\frac{3p}{8}\right) \right\}$$

si  $f=3$

$$H(4) = \frac{1}{2} \left\{ H_0(4) + H_1(4)\cos\left(\frac{p}{2}\right) + H_1(4)\sen\left(\frac{p}{2}\right) \right\}$$

$$H(12) = \frac{1}{2} \left\{ H_0(4) - H_1(4)\cos\left(\frac{p}{2}\right) - H_1(4)\sen\left(\frac{p}{2}\right) \right\}$$

si  $f=4$

$$H(5) = \frac{1}{2} \left\{ H_0(5) + H_1(5) \cos\left(\frac{5p}{8}\right) + H_1(3) \operatorname{sen}\left(\frac{5p}{8}\right) \right\}$$

$$H(13) = \frac{1}{2} \left\{ H_0(5) - H_1(5) \cos\left(\frac{5p}{8}\right) - H_1(3) \operatorname{sen}\left(\frac{5p}{8}\right) \right\} \quad \text{si } f=5$$

$$H(6) = \frac{1}{2} \left\{ H_0(6) + H_1(6) \cos\left(\frac{3p}{4}\right) + H_1(2) \operatorname{sen}\left(\frac{3p}{4}\right) \right\}$$

$$H(14) = \frac{1}{2} \left\{ H_0(6) - H_1(6) \cos\left(\frac{3p}{4}\right) - H_1(2) \operatorname{sen}\left(\frac{3p}{4}\right) \right\} \quad \text{si } f=6$$

$$H(7) = \frac{1}{2} \left\{ H_0(7) + H_1(7) \cos\left(\frac{7p}{8}\right) + H_1(1) \operatorname{sen}\left(\frac{7p}{8}\right) \right\}$$

$$H(15) = \frac{1}{2} \left\{ H_0(7) - H_1(7) \cos\left(\frac{7p}{8}\right) - H_1(1) \operatorname{sen}\left(\frac{7p}{8}\right) \right\} \quad \text{si } f=7$$

Este proceso recursivo continúa hasta la primera columna, en la cual se aprecian 16 transformadas de Hartley de longitud 1, cuyo resultado son los mismos valores. Por lo tanto, la primera columna representa los valores de la secuencia original en el tiempo  $X(t)$ . El orden extraño en el cual se encuentran se explica por el *bit de orden inverso*, el cual se explicó ya en el apartado anterior.

#### 5.3.2.1. Implementación:

Para realizar una implementación correcta del algoritmo de la transformada de Hartley, hay que tener en cuenta que se debe disponer de 2 vectores, uno de origen y el otro de destino. Esto debido a que para realizar los cálculos en las mariposas se requiere tres entradas para obtener dos salidas. Si el resultado de la mariposa se sobrescribiera en los datos, otro cálculo posterior podría tener un resultado errado porque el dato original que había se cambió por otro.

En general, cada conjunto de cálculos requiere  $N/2$  evaluaciones en las ecuaciones definitivas.

Cada evaluación requiere la realización de varias operaciones como son la evaluación de funciones seno y coseno, dos multiplicaciones y dos sumas. Para elevar la velocidad de cómputo y disminuir el tamaño de programa, se optó por usar tablas de datos con los valores de seno y coseno ya realizadas; esto se conoce como "*pre-computed lookup table*". En el programa se implementó una tabla de 256 datos evaluada para valores de ángulo en radianes correspondientes



al argumento de las funciones trigonométricas:  $q = \frac{2pf}{N}$ . Si se va a realizar el programa para

$N=1024$ :

$$q = \frac{2pf}{N} = \frac{2pf}{1024} = \frac{pf}{512}$$

Esto daría lugar a una tabla de 512 posiciones, pero se pueden aprovechar las identidades trigonométricas para reducir aún más esta tabla:

$$\sin\left(x + \frac{p}{2}\right) = \sin(x)$$

$$\cos(x) = \sin\left(\frac{p}{2} - x\right)$$

$$\cos\left(x + \frac{p}{2}\right) = -\cos(x)$$

Aprovechando la periodicidad de las funciones seno y coseno, se construyó una tabla de 256 posiciones, la cual abarca todos los ángulos necesarios. El programa, como se verá más adelante, se encarga de hallar el valor correspondiente que se necesita según el valor de  $f$  y de  $N$  en el cual se halle en el momento. La siguiente es la tabla que se usó para este fin (tabla 2):

f	t=f*pi/512	coseno(t)	seno(t)	f	t=f*pi/512	coseno(t)	seno(t)
0	0.0000	1.0000	0.0000	129	0.7915	0.7028	0.7114
1	0.0061	1.0000	0.0061	130	0.7977	0.6984	0.7157
2	0.0123	0.9999	0.0123	131	0.8038	0.6940	0.7200
3	0.0184	0.9998	0.0184	132	0.8099	0.6895	0.7242
4	0.0245	0.9997	0.0245	133	0.8161	0.6851	0.7285
5	0.0307	0.9995	0.0307	134	0.8222	0.6806	0.7327
6	0.0368	0.9993	0.0368	135	0.8283	0.6761	0.7368
7	0.0430	0.9991	0.0429	136	0.8345	0.6716	0.7410
8	0.0491	0.9988	0.0491	137	0.8406	0.6670	0.7451
9	0.0552	0.9985	0.0552	138	0.8468	0.6624	0.7491
10	0.0614	0.9981	0.0613	139	0.8529	0.6578	0.7532
11	0.0675	0.9977	0.0674	140	0.8590	0.6532	0.7572
12	0.0736	0.9973	0.0736	141	0.8652	0.6485	0.7612

13	0.0798	0.9968	0.0797	142	0.8713	0.6438	0.7652
14	0.0859	0.9963	0.0858	143	0.8774	0.6391	0.7691
15	0.0920	0.9958	0.0919	144	0.8836	0.6344	0.7730
16	0.0982	0.9952	0.0980	145	0.8897	0.6296	0.7769
17	0.1043	0.9946	0.1041	146	0.8958	0.6249	0.7807
18	0.1104	0.9939	0.1102	147	0.9020	0.6201	0.7846
19	0.1166	0.9932	0.1163	148	0.9081	0.6152	0.7883
20	0.1227	0.9925	0.1224	149	0.9143	0.6104	0.7921
21	0.1289	0.9917	0.1285	150	0.9204	0.6055	0.7958
22	0.1350	0.9909	0.1346	151	0.9265	0.6006	0.7995
23	0.1411	0.9901	0.1407	152	0.9327	0.5957	0.8032
24	0.1473	0.9892	0.1467	153	0.9388	0.5908	0.8068
25	0.1534	0.9883	0.1528	154	0.9449	0.5858	0.8105
26	0.1595	0.9873	0.1589	155	0.9511	0.5808	0.8140
27	0.1657	0.9863	0.1649	156	0.9572	0.5758	0.8176
28	0.1718	0.9853	0.1710	157	0.9633	0.5708	0.8211
29	0.1779	0.9842	0.1770	158	0.9695	0.5657	0.8246
30	0.1841	0.9831	0.1830	159	0.9756	0.5607	0.8280
31	0.1902	0.9820	0.1891	160	0.9817	0.5556	0.8315
32	0.1963	0.9808	0.1951	161	0.9879	0.5505	0.8349
33	0.2025	0.9796	0.2011	162	0.9940	0.5453	0.8382
34	0.2086	0.9783	0.2071	163	1.0002	0.5402	0.8416
35	0.2148	0.9770	0.2131	164	1.0063	0.5350	0.8449
36	0.2209	0.9757	0.2191	165	1.0124	0.5298	0.8481
37	0.2270	0.9743	0.2251	166	1.0186	0.5246	0.8514
38	0.2332	0.9729	0.2311	167	1.0247	0.5194	0.8546
39	0.2393	0.9715	0.2370	168	1.0308	0.5141	0.8577
40	0.2454	0.9700	0.2430	169	1.0370	0.5088	0.8609
41	0.2516	0.9685	0.2489	170	1.0431	0.5035	0.8640
42	0.2577	0.9670	0.2549	171	1.0492	0.4982	0.8670
43	0.2638	0.9654	0.2608	172	1.0554	0.4929	0.8701
44	0.2700	0.9638	0.2667	173	1.0615	0.4876	0.8731
45	0.2761	0.9621	0.2726	174	1.0677	0.4822	0.8761
46	0.2823	0.9604	0.2785	175	1.0738	0.4768	0.8790
47	0.2884	0.9587	0.2844	176	1.0799	0.4714	0.8819
48	0.2945	0.9569	0.2903	177	1.0861	0.4660	0.8848
49	0.3007	0.9551	0.2962	178	1.0922	0.4605	0.8876

50	0.3068	0.9533	0.3020	179	1.0983	0.4551	0.8904
51	0.3129	0.9514	0.3078	180	1.1045	0.4496	0.8932
52	0.3191	0.9495	0.3137	181	1.1106	0.4441	0.8960
53	0.3252	0.9476	0.3195	182	1.1167	0.4386	0.8987
54	0.3313	0.9456	0.3253	183	1.1229	0.4331	0.9013
55	0.3375	0.9436	0.3311	184	1.1290	0.4276	0.9040
56	0.3436	0.9415	0.3369	185	1.1351	0.4220	0.9066
57	0.3497	0.9395	0.3427	186	1.1413	0.4164	0.9092
58	0.3559	0.9373	0.3484	187	1.1474	0.4108	0.9117
59	0.3620	0.9352	0.3542	188	1.1536	0.4052	0.9142
60	0.3682	0.9330	0.3599	189	1.1597	0.3996	0.9167
61	0.3743	0.9308	0.3656	190	1.1658	0.3940	0.9191
62	0.3804	0.9285	0.3713	191	1.1720	0.3883	0.9215
63	0.3866	0.9262	0.3770	192	1.1781	0.3827	0.9239
64	0.3927	0.9239	0.3827				
65	0.3988	0.9215	0.3883	193	1.1842	0.3770	0.9262
66	0.4050	0.9191	0.3940	194	1.1904	0.3713	0.9285
67	0.4111	0.9167	0.3996	195	1.1965	0.3656	0.9308
68	0.4172	0.9142	0.4052	196	1.2026	0.3599	0.9330
69	0.4234	0.9117	0.4108	197	1.2088	0.3542	0.9352
70	0.4295	0.9092	0.4164	198	1.2149	0.3484	0.9373
71	0.4357	0.9066	0.4220	199	1.2210	0.3427	0.9395
72	0.4418	0.9040	0.4276	200	1.2272	0.3369	0.9415
73	0.4479	0.9013	0.4331	201	1.2333	0.3311	0.9436
74	0.4541	0.8987	0.4386	202	1.2395	0.3253	0.9456
75	0.4602	0.8960	0.4441	203	1.2456	0.3195	0.9476
76	0.4663	0.8932	0.4496	204	1.2517	0.3137	0.9495
77	0.4725	0.8904	0.4551	205	1.2579	0.3078	0.9514
78	0.4786	0.8876	0.4605	206	1.2640	0.3020	0.9533
79	0.4847	0.8848	0.4660	207	1.2701	0.2962	0.9551
80	0.4909	0.8819	0.4714	208	1.2763	0.2903	0.9569
81	0.4970	0.8790	0.4768	209	1.2824	0.2844	0.9587
82	0.5031	0.8761	0.4822	210	1.2885	0.2785	0.9604
83	0.5093	0.8731	0.4876	211	1.2947	0.2726	0.9621
84	0.5154	0.8701	0.4929	212	1.3008	0.2667	0.9638
85	0.5216	0.8670	0.4982	213	1.3070	0.2608	0.9654
86	0.5277	0.8640	0.5035	214	1.3131	0.2549	0.9670

87	0.5338	0.8609	0.5088	215	1.3192	0.2489	0.9685
88	0.5400	0.8577	0.5141	216	1.3254	0.2430	0.9700
89	0.5461	0.8546	0.5194	217	1.3315	0.2370	0.9715
90	0.5522	0.8514	0.5246	218	1.3376	0.2311	0.9729
91	0.5584	0.8481	0.5298	219	1.3438	0.2251	0.9743
92	0.5645	0.8449	0.5350	220	1.3499	0.2191	0.9757
93	0.5706	0.8416	0.5402	221	1.3560	0.2131	0.9770
94	0.5768	0.8382	0.5453	222	1.3622	0.2071	0.9783
95	0.5829	0.8349	0.5505	223	1.3683	0.2011	0.9796
96	0.5890	0.8315	0.5556	224	1.3744	0.1951	0.9808
97	0.5952	0.8280	0.5607	225	1.3806	0.1891	0.9820
98	0.6013	0.8246	0.5657	226	1.3867	0.1830	0.9831
99	0.6075	0.8211	0.5708	227	1.3929	0.1770	0.9842
100	0.6136	0.8176	0.5758	228	1.3990	0.1710	0.9853
101	0.6197	0.8140	0.5808	229	1.4051	0.1649	0.9863
102	0.6259	0.8105	0.5858	230	1.4113	0.1589	0.9873
103	0.6320	0.8068	0.5908	231	1.4174	0.1528	0.9883
104	0.6381	0.8032	0.5957	232	1.4235	0.1467	0.9892
105	0.6443	0.7995	0.6006	233	1.4297	0.1407	0.9901
106	0.6504	0.7958	0.6055	234	1.4358	0.1346	0.9909
107	0.6565	0.7921	0.6104	235	1.4419	0.1285	0.9917
108	0.6627	0.7883	0.6152	236	1.4481	0.1224	0.9925
109	0.6688	0.7846	0.6201	237	1.4542	0.1163	0.9932
110	0.6750	0.7807	0.6249	238	1.4603	0.1102	0.9939
111	0.6811	0.7769	0.6296	239	1.4665	0.1041	0.9946
112	0.6872	0.7730	0.6344	240	1.4726	0.0980	0.9952
113	0.6934	0.7691	0.6391	241	1.4788	0.0919	0.9958
114	0.6995	0.7652	0.6438	242	1.4849	0.0858	0.9963
115	0.7056	0.7612	0.6485	243	1.4910	0.0797	0.9968
116	0.7118	0.7572	0.6532	244	1.4972	0.0736	0.9973
117	0.7179	0.7532	0.6578	245	1.5033	0.0674	0.9977
118	0.7240	0.7491	0.6624	246	1.5094	0.0613	0.9981
119	0.7302	0.7451	0.6670	247	1.5156	0.0552	0.9985
120	0.7363	0.7410	0.6716	248	1.5217	0.0491	0.9988
121	0.7424	0.7368	0.6761	249	1.5278	0.0429	0.9991
122	0.7486	0.7327	0.6806	250	1.5340	0.0368	0.9993
123	0.7547	0.7285	0.6851	251	1.5401	0.0307	0.9995

124	0.7609	0.7242	0.6895	252	1.5463	0.0245	0.9997
125	0.7670	0.7200	0.6940	253	1.5524	0.0184	0.9998
126	0.7731	0.7157	0.6984	254	1.5585	0.0123	0.9999
127	0.7793	0.7114	0.7028	255	1.5647	0.0061	1.0000
128	0.7854	0.7071	0.7071	256	1.5708	0.0000	1.0000

**Tabla 2.** Tabla con los valores de seno y coseno para ángulos entre 0 grados y 90 grados

La interacción entre el programa y la tabla se expone más adelante.

#### 5.3.2.2. Ventajas de la Transformada de Hartley:

- Para ser implementada requiere de menos multiplicaciones y sumas que la Transformada Rápida de Fourier. Es hasta 3 veces más rápida que una FFT porque no requiere realizar operaciones complejas (como referencia, ver en la bibliografía documento de OTOLLINI, *Migration by Hartley Transform*). Por lo tanto, su tiempo de cómputo es mucho menor. Como referencia, se tiene que la complejidad computacional  $O$  de una Transformada de Fourier Discreta es del orden de  $O(N^2)$ , de una Transformada Rápida de Fourier es de  $O(N \log N)$  y de una transformada de Hartley es de  $O((N \log N)/3)$
- La transformada de Hartley requiere de menos memoria para almacenar números que la Transformada Rápida de Fourier, porque no usa números complejos. Los resultados de la transformada de Hartley se almacenan en un vector que tiene el mismo número de componentes que el vector de datos inicial.

#### 5.3.2.3. Desventajas de la Transformada de Hartley:

- Su única desventaja se fundamenta en la aproximación que se realiza al eliminar la parte compleja de la definición de la Transformada de Fourier. En aplicaciones donde considerar la parte compleja es importante, no se debe usar la transformada de Hartley.

### 5.4. Análisis espectral del la señal ECG

Una señal ECG, como cualquier señal, se puede descomponer en una gran variedad de señales en varias frecuencias. Si se consigue obtener el espectro de una señal ECG, se podrán identificar diversas patologías en el sistema cardíaco. A continuación se expone la forma como se puede obtener el espectro de una señal a partir de su transformada de Hartley.

Aunque la función de densidad espectral de una señal se defina como la transformada de Fourier de su función de autocorrelación, es posible demostrar que no es necesario calcular la función de autocorrelación para obtener la función de densidad espectral, ya que esta función puede estimarse de forma directa a partir de la serie muestreada con anterioridad. Una demostración de la anterior afirmación se puede encontrar en los sitios web de E. Barrull (apéndice B – C) y el de Athanasios Papoulis (diapositiva 18). Ver la bibliografía para mayor información.

En series discretas, la función de densidad espectral  $S_K$ , donde el subíndice  $K$  determina la frecuencia asociada  $w_k = \frac{2\pi k}{N\Delta}$ , siendo  $\Delta$  el intervalo de muestreo y  $N$  el número de muestras.

$S_K$  viene dado por:

$$S(k) = X_K^* X_K = |X_K|^2$$

Donde  $X_K$  es la DFT  $k$ -ésima de la serie discreta  $\{x_r\}$  y  $X_K^*$  es su compleja conjugada. De esta manera no es necesario hallar la función de autocorrelación de la serie  $\{x_r\}$  para obtener la densidad espectral, basta el cálculo de la DFT.

Ésta es la función que se utiliza para obtener el espectro de la señal ECG, el cual se grafica en la pantalla de cristal líquido.