

515.723
Av 784
552
T-E2

ESCUELA POLITECNICA NACIONAL
FACULTAD DE INGENIERIA ELECTRICA

" T R A N S F O R M A D A R A P I D A C O S E N O "

TESIS PREVIA A LA OBTENCION DEL TITULO
DE INGENIERO EN LA ESPECIALIZACION DE
ELECTRONICA Y TELECOMUNICACIONES

JAIME ROLANDO ARTURO ERAZO

QUITO, AGOSTO 1983



Certifico que el presente
trabajo ha sido elaborado
en su totalidad por Jaime
Arturo Erazo.

A handwritten signature in black ink, appearing to read 'G. Hidalgo', written over a horizontal line.

Ing. Gualberto Hidalgo

Director de tesis

DEDICATORIA

A Guillermo Efraín y Cruz Helena, mis padres, por ese apoyo, cariño y comprensión que siempre me han brindado para ellos mi eterno agradecimiento.

A María Helena, Carmen Alicia y Ana Mercedes, mis hermanas, les adeudo mi gratitud de su constante apoyo.

AGRADECIMIENTO

A mi director de tesis Ing. Gualberto Hidalgo por sus oportunas sugerencias y ayuda prestada en la consecución de el presente trabajo; al Ing. Efraín del Pino por su amabilidad y colaboración en el computador; también a mis amigos por su colaboración desinteresada.

Finalmente mi agradecimiento especial a la Facultad de Ingeniería Eléctrica a la Escuela Politécnica Nacional y a Ecuador de los que conservo gratos recuerdos.

I N D I C E

INTRODUCCION	4
1. TRANSFORMADA DISCRETA DE FOURIER	
1.1 TRANSFORMADA DE FOURIER	7
1.2 TRANSFORMADA RAPIDA DE FOURIER	13
2. TRANSFORMADA RAPIDA COSENO	
2.1 TRANSFORMADA DISCRETA COSENO	25
2.2 TRANSFORMADA DISCRETA COSENO DIRECTA	28
2.3 TRANSFORMADA DISCRETA COSENO INVERSA	33
2.4 TRANSFORMADA RAPIDA COSENO DIRECTA	37
2.5 TRANSFORMADA RAPIDA COSENO INVERSA	41
3. TRANSFORMADA OPTIMA DE KARHUNEN LOEVE	
3.1 TRANSFORMADA OPTIMA DE KARHUNEN LOEVE	47
4. ALGORITMOS	
4.1 ALGORITMO PARA EL CALCULO DE LA TRANSFORMADA RAPIDA COSENO	51
4.2 ALGORITMO PARA EL CALCULO DE LA TRANSFORMADA DE KARHUNEN LOEVE	60
5. FILTRADO DE WIENER	
5.1 SOLUCION POR LA TRANSFORMADA RAPIDA COSENO	68
5.2 OTRAS SOLUCIONES	71

El tema principal en esta tesis trata sobre la Transformada Coseno, implementada con un algoritmo de cálculo rápido con el que se puede trabajar en aplicaciones que requieran resultados en el menor tiempo posible.

Se logra este objetivo con el concurso de la transformada Rápida de Fourier la misma que se encuentra resumida en el primer capítulo.

El método consiste en alimentar la Transformada Rápida de Fourier con un vector cuya longitud es la mitad de la del vector de entrada y que contiene toda la información. Notemos que los datos en la Transformada Coseno son reales y que en la Transformada Rápida de Fourier pueden ser complejos; en esta forma se pueden generar todos los resultados de los datos como veremos en los capítulos 2 y 4.

La aplicación que se da a este algoritmo es en el Filtrado de Wiener. Este tipo de filtrado da los mejores resultados con la Transformada de Karhunen Loeve, la que tratamos de manera informativa.

Al final, después del capítulo dedicado a los algoritmos, se presentan los aspectos más importantes del filtrado, los resultados y las conclusiones.

1.1 TRANSFORMADA DE FOURIER

En este capítulo veremos como se consigue adaptar una transformada de funciones continuas para ser manejadas por el computador. La transformada continua de Fourier de la forma [1]:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (1.1)$$

y su inversa

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \quad (1.2)$$

tienen los límites entre $-\infty$ y $+\infty$, y aunque no todas las señales pueden tener esa extensión no es práctico que ésta sea demasiado grande, por esta razón generalmente nos limitamos a un intervalo finito. Adicionalmente, por tratarse de funciones continuas, no pueden introducirse directamente al computador; para este fin utilizamos (basados en el teorema del muestreo) valores discretos de la señal a intervalos equidistantes con una frecuencia igual o mayor a dos veces la frecuencia de la componente más alta de la se-
ñal.

Para visualizar rápidamente este proceso tomemos una señal cualquie-

ra como la de la figura 1.1, que representa una señal continua.

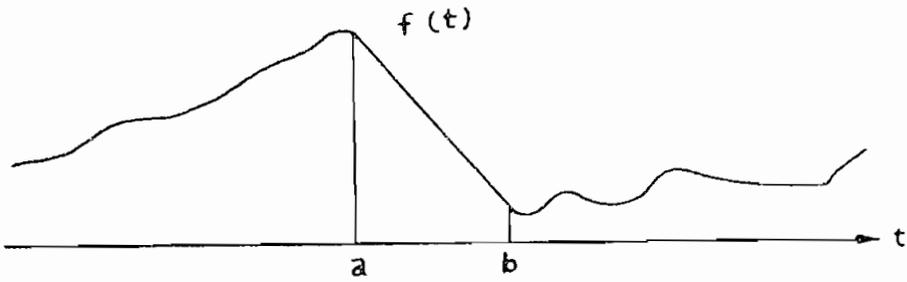


Fig. 1.1.

Limitándonos al intervalo $[a, b]$ que es donde la señal nos interesa, podemos elegir nuestro sistema de coordenadas como en la figura 1.2, de tal manera que la señal comience en el origen de coordenadas, y, en el caso que tomemos N - muestras, el fin de la señal lo ubicaremos en $N - 1$.

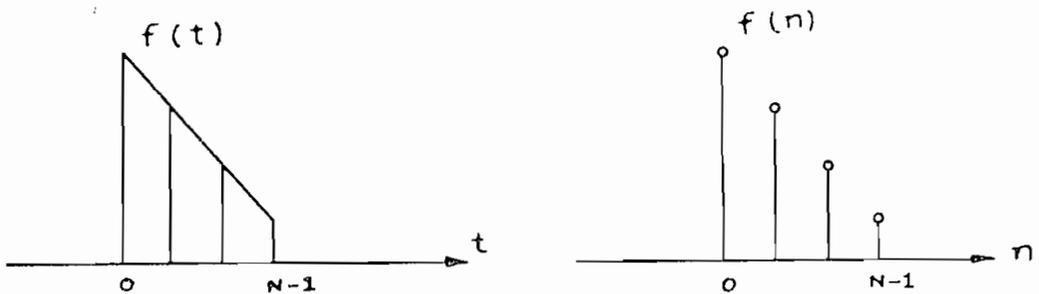


Fig. 1.2.

En estas condiciones, todos los valores pasan a ser discretos, nuestra integral a un sumatorio de un número determinado de valores y las diferenciales a intervalos.

Así (1.1) y (1.2) se escriben de la siguiente manera:

$$F(\omega) = \sum_{t=0}^{N-1} f(t) e^{-j\omega t} \Delta t \quad (1.3)$$

$$f(t) = \frac{1}{2\pi} \sum_{\omega=-N/2}^{N/2} F(\omega) e^{j\omega t} \Delta \omega, \quad (1.4)$$

En este momento hacemos un cambio de variables para diferenciar la transformada discreta que queremos obtener de la continua, para esto hacemos n igual a los momentos (discretos) en que muestreemos la señal así:

$$n = t$$

$$n = 0, 1, \dots, N; \text{ Unidades de tiempo}$$

Δt será el intervalo total de la señal entre el número de muestras menos una.

$$\Delta t = \frac{(N-1) - 0}{N - 1} = 1$$

La máxima frecuencia estará determinada por Δt

$$W_{\max} = \frac{2 \pi}{\Delta t} = 2 \pi$$

De tal manera que:

$$\Delta w = \frac{w_{\max}}{N} = \frac{2 \pi}{N}$$

y w será:

$$w = \frac{2 \pi}{N} k, \quad -\frac{N}{2} \leq k \leq \frac{N}{2}$$

En estas condiciones de las variables, w se relaciona con k y t con n . Ahora si reescribimos (1.3) y (1.4) obtenemos:

$$F(k) = \sum_{n=0}^{N-1} f(n) e^{-j \frac{2 \pi}{N} kn} \quad (1) \quad (1.5)$$

$$f(n) = \frac{1}{2\pi} \sum_{k=-N/2}^{N/2} F(k) e^{j \frac{2\pi}{N} kn} \left(\frac{2\pi}{N} \right), \quad (1.6)$$

Reordenando términos estas ecuaciones toman la forma:

$$F(k) = \sum_{n=0}^{N-1} f(n) W_N^{nk} \quad (1.7)$$

$$f(n) = \frac{1}{N} \sum_{k=-N/2}^{N/2} F(k) W_N^{-nk} \quad (1.8)$$

Donde se introduce la notación:

$$e^{-j \frac{2\pi}{N} kn} = W_N^{kn}$$

Podemos escribir la transformada discreta de Fourier en la forma matricial:

$$\begin{bmatrix} F(k) \end{bmatrix} = \begin{bmatrix} W_N^{nk} \\ W_N^{nk} \\ \vdots \\ W_N^{nk} \end{bmatrix} \begin{bmatrix} f(n) \end{bmatrix} \quad (1.9)$$

$$\begin{bmatrix} f(n) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} W_N^{-nk} \\ W_N^{-nk} \\ \vdots \\ W_N^{-nk} \end{bmatrix} \begin{bmatrix} F(k) \end{bmatrix} \quad (1.10)$$

Donde:

$$\begin{bmatrix} f(n) \end{bmatrix}, \begin{bmatrix} F(k) \end{bmatrix}$$

son matrices columnas de $N \times 1$ elementos y

$$\begin{bmatrix} W & -nk \\ N \end{bmatrix}, \begin{bmatrix} W & nk \\ N \end{bmatrix}$$

son matrices cuadradas de $N \times N$ elementos. Estas son las matrices de transformación o de Fourier, las que son fáciles de factorizar para obtener mayor rendimiento computacional.

Hay un método rápido para calcular las matrices factor directamente, en el cual se basa la transformada rápida de Fourier que revisamos enseguida.

iii) se pasa nuevamente a decimal, obteniéndose así $\langle k \rangle$

La segunda matriz factor se halla haciendo el producto de Kronecker de las $N/4$ - primeras submatrices con $\begin{bmatrix} I_2 \end{bmatrix}$ que forman la diagonal principal y el resto se completa con ceros.

La tercera matriz factor se obtiene en forma similar a la segunda pero con las $\frac{N}{8}$ - primeras submatrices de la segunda matriz factor.

La p -ésima matriz será el producto de Kronecker entre las $\frac{N}{2^p}$ - primeras submatrices de la $(p-1)$ - ésima matriz y la matriz identidad $\begin{bmatrix} I \\ 2 \end{bmatrix}$.

Así llegamos a conseguir las $(\log_2 N)$ - matrices factor.

Para aclarar, la forma de obtener las matrices factor desarrollamos el procedimiento para el caso en que $N = 16$

En primer lugar obtenemos: k

$$K = \log_2 \frac{N}{2}$$

$$K = \log_2 \frac{16}{2}$$

$$\begin{bmatrix}
 1 & w_{16}^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1-w_{16}^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & w_{16}^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1-w_{16}^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & w_{16}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1-w_{16}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & w_{16}^6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1-w_{16}^6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & w_{16}^1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-w_{16}^1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & w_{16}^5 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-w_{16}^5 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & w_{16}^3 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-w_{16}^3 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & w_{16}^7 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-w_{16}^7 & 16
 \end{bmatrix}$$

La segunda matriz factor, será el producto de Kronecker de las $(N/4)$,
4 primeras submatrices con $[I_2]$

En general :

$$\begin{bmatrix} 1 & W_{16}^k \\ 1 & -W_{16}^k \end{bmatrix} \oplus \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{matrix} 18 \\ \begin{bmatrix} 1 & 0 & W_{16}^k & 0 \\ 0 & 1 & 0 & -W_{16}^k \\ 1 & 0 & -W_{16}^k & 0 \\ 0 & 1 & 0 & -W_{16}^k \end{bmatrix} \end{matrix}$$

Obteniéndose para la segunda Matriz:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -j & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -j & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & j & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & j & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & W_{16}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & W_{16}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -W_{16}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -W_{16}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & W_{16}^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & W_{16}^6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -W_{16}^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -W_{16}^6 \end{bmatrix}$$

Donde se han utilizado las siguientes equivalencias:

$$W_{16}^0 = 1; \quad W_{16}^4 = -j$$

La tercera matriz factor, será el producto de Kronecker de las (N/8),
2 primeras submatrices de la matriz anterior con I_2 , así:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \oplus \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Con la segunda submatriz tenemos:

$$\begin{bmatrix} 1 & 0 & -j & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & j & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \oplus \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & -j & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -j & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -j & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -j \\ -j & 0 & 0 & 0 & j & 0 & 0 & 0 \\ 0 & -j & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & -j & 0 & 0 & 0 & j & 0 \\ 0 & 0 & 0 & -j & 0 & 0 & 0 & j \end{bmatrix}$$

Resultando así la tercera matriz factor.

La figura 1.3 muestra las cuatro matrices factor para el caso cuando el número de muestras son $N = 16$.

Observando las matrices factor, notamos que contienen un número significativo de elementos ceros; éstos son los que facilitan los cálculos computacionales en la Transformada Rápida de Fourier, que utilizamos como base para la transformada rápida coseno que estudiamos en el siguiente capítulo.

2.1. TRANSFORMADA DISCRETA COSENO

Sea $f(n)$ una señal discreta en el tiempo y $F(k)$ su transformada de Fourier; es decir: $f(n)$ cumple con la ecuación (1.7).

Por definición la transformada coseno de una señal, es la parte real de la transformada de Fourier de dicha señal.

Por otro lado, la parte real es la transformada de la parte par de la señal para $f(n)$ definida, como: [2]

$$f_p(n) = \left[f(n) + f(-n) \right] / 2 \quad (2.1)$$

Si $f(n)$ es causal; es decir; $f(n) = 0$ para todo $n < 0$ entonces $f(n)$ y su transformada coseno son especificadas en forma única. Así, $f_p(n)$ puede pensarse como una extensión par de $f(n)$. En consecuencia la transformada coseno puede obtenerse como la transformada de Fourier de la extensión par de $f(n)$ causal. Para este caso construimos nuestra señal de acuerdo a nuestros requerimientos.

Como apreciamos en el capítulo anterior, el cálculo usual de la transformada discreta de Fourier se realiza en un número finito de muestras a intervalos de igual frecuencia.



Para este caso la señal puede ser recuperada de la transformada discreta de Fourier en su forma periódica.

Respecto a extensiones par de una señal donde se obliga a esta a ser periódica uno puede elegir estas extensiones.

Existen diferentes maneras de lograr extensiones par de una señal de N muestras con períodos $(2N - 2)$, $(2N - 1)$ y $2N$ muestras, las cuales sirven de base para la definición de transformada discreta - coseno. En el presente trabajo sólo discutiremos extensiones par de $2N -$ puntos.

Para visualizar veremos el siguiente ejemplo:

La figura 2.1 representa nuestra señal causal $f(n)$.

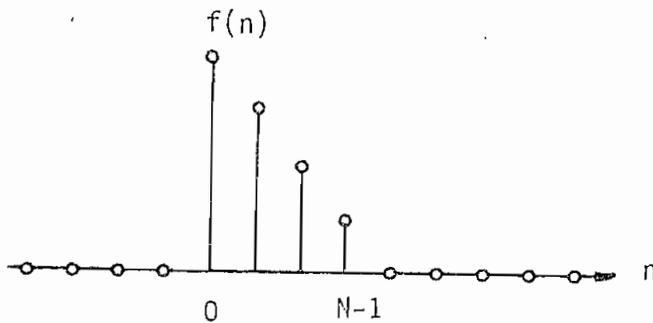


Fig. 2.1

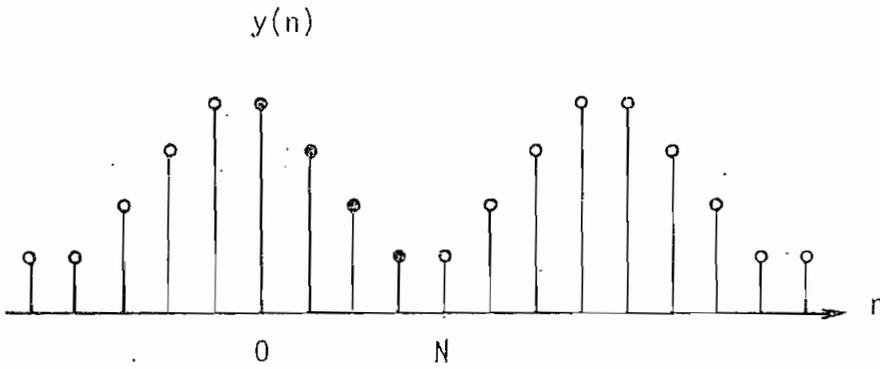


Fig. 2.2

La figura 2.2 nos muestra la extensión par de $2N$ - puntos de la señal, la que es par alrededor del punto $-0,5$. La transformada de Fourier de esta extensión contiene un término con defasaje lineal, correspondiente a media muestra.

Esta es la extensión que utilizaremos en adelante.

2.2. TRANSFORMADA DISCRETA COSENO DIRECTA

Teniendo $f(n)$, una secuencia de N - puntos en el intervalo $[0, N-1]$, deseamos su transformada directa coseno, derivada con la ayuda de la transformada de Fourier de una extensión par de $f(n)$ de $2N$ -puntos.

Sea $y(n)$ una extensión par de $2N$ -puntos de $f(n)$ definida por:

$$y(n) = \begin{cases} f(n), & 0 \leq n \leq N-1 \\ f(2N-n-1), & N \leq n \leq 2N-1 \end{cases} \quad (2.2)$$

Entonces:

$$y(2N-n-1) = y(n) \quad (2.3)$$

En el ejemplo; tenemos en la figura 2.1, la señal $f(n)$ y en la figura 2.2 la correspondiente extensión par $y(n)$ definida como en (2.2). En la ecuación (2.3) podemos observar que hay un -1 en el primer miembro de la ecuación, que es el que no permite a la extensión ser verdaderamente par respecto al origen, por consiguiente no tenemos una transformada discreta Fourier real, como vemos en seguida.

La transformada discreta de Fourier de $f(n)$ está dada como en (1.7) por:

$$Y(k) = \sum_{n=0}^{2N-1} f(n) W^{nk} \quad (2.4)$$

Sustituyendo (2.2) en (2.4) tenemos:

$$Y(k) = \sum_{n=0}^{N-1} f(n) w_{2N}^{nk} + \sum_{n=N}^{2N-1} f(2N-n-1) w_{2N}^{nk} \quad (2.5)$$

Cambiando variable tenemos:

$$2N - n - 1 = n'$$

$$-n = n' - 2N + 1$$

$$n = 2N - n' - 1$$

Cuando:

$$n = N; n' = 2N - N - 1 = N - 1$$

$$n = 2N - 1; n' = 2N - (2N - 1) - 1 = 0$$

Así tenemos:

$$Y(k) = \sum_{n=0}^{N-1} f(n) w_{2N}^{nk} + \sum_{n'=0}^{N-1} f(n') w_{2N}^{(2N-n'-1)k} \quad (2.6)$$

Hacemos $n' = n$

$$Y(k) = \sum_{n=0}^{N-1} f(n) W_{2N}^{nk} + \sum_{n=0}^{N-1} f(n) W_{2N}^{2Nk} W_{2N}^{-nk} W_{2N}^{-k} \quad (2.7)$$

Veamos que es:

$$W_{2N}^{2Nk}$$

$$W_{2N}^{2Nk} = e^{-j \frac{2\pi \cdot 2Nk}{2N}} = e^{-j2\pi k} = 1 \quad (2.8)$$

Factorizando $W_{2N}^{-k/2}$ nos queda:

$$Y(k) = W_{2N}^{-k/2} \sum_{n=0}^{N-1} f(n) \left[W_{2N}^{2nk} W_{2N}^{k/2} + W_{2N}^{-nk} W_{2N}^{-k/2} \right] \quad (2.9)$$

La expresión (2.9) podemos describirla de dos maneras:

$$Y(k) = W_{2N}^{-k/2} \sum_{n=0}^{N-1} f(n) \cos \frac{(2n+1)\pi k}{2N}, \quad 0 \leq k \leq 2N-1 \quad (2.10)$$

o :

$$Y(k) = W_{2N}^{-k/2} 2 \operatorname{Re} \left[W_{2N}^{k/2} \sum_{n=0}^{N-1} f(n) W_{2N}^{nk} \right], \quad 0 \leq k \leq 2N-1 \quad (2.11)$$

Notemos de $C(k)$ que es real y $Y(k)$ compleja, además, $Y(k)$ puede llegar a ser $C(k)$ retrasando la secuencia $y(n)$ media muestra, en otras palabras con este retraso lograríamos que $y(n)$ sea verdaderamente par.

En consecuencia la transformada discreta coseno puede obtenerse - calculando la transformada discreta de Fourier de la secuencia de $2N$ puntos $y(n)$ y multiplicando el resultado por $W^{k/2}$ como en (2.14).

Si observamos (2.15) podemos obtener también el resultado tomando dos veces la parte real de multiplicar por $W^{k/2}$ la Transformada de Fourier de la secuencia original $f(n)$ añadida N ceros. [4]

2.3. TRANSFORMADA DISCRETA COSENO INVERSA

De igual manera que la directa podemos derivar la transformada discreta coseno inversa de la transformada discreta de Fourier inversa. Partamos de (1.8) y con $y(n) = f(n)$.

$$y(n) = \frac{1}{2N} \sum_{k=0}^{2N-1} Y(k) W^{\frac{-nk}{2N}} \quad (2.16)$$

Como $y(n)$ es real, $Y(k)$ es hermiciana simétrica, es decir cumple con:

$$Y(2N-k) = Y^*(k) \quad (2.17)$$

Para $Y(N)$ de (2.9) tenemos:

$$Y(N) = W^{\frac{-N/2}{2N}} \sum_{n=0}^{N-1} y(n) \left[W^{\frac{nN}{2N}} W^{\frac{N/2}{2N}} + W^{\frac{-nN}{2N}} W^{\frac{-N/2}{2N}} \right]$$

$$Y(N) = W^{\frac{-N/2}{2N}} \sum_{n=0}^{N-1} y(n) \left[W^{\frac{2n+1}{4}} + W^{\frac{-(2n+1)}{4}} \right]$$

$$Y(N) = W^{\frac{-1}{4}} \sum_{n=0}^{N-1} y(n) \left[\cos \frac{\pi}{2} (2n+1) - j \operatorname{Sen} \frac{\pi}{2} (2n+1) + \cos \frac{\pi}{2} (2n+1) + j \operatorname{Sen} \frac{\pi}{2} (2n+1) \right]$$

$$Y(N) = \frac{1}{4} \sum_{n=0}^{N-1} y(n) \cdot 2 \cos \frac{\pi}{2} (2n+1)$$

Para n entero $\cos \frac{\pi}{2} (2n+1) = 0$, entonces

$$Y(N) = 0 \quad (2.18)$$

Introduciendo (2.17) y (2.18) en (2.16) tenemos:

$$y(n) = \frac{1}{2N} \sum_{k=0}^{2N-1} Y(k) W_{2N}^{-nk}$$

$$y(n) = \frac{1}{2N} \left[\sum_{k=0}^{N-1} Y(k) W_{2N}^{-nk} + \sum_{k=1}^N Y(2N-k) W_{2N}^{-n(2N-k)} \right]$$

$$y(n) = \frac{1}{2N} \left[Y(0) W_{2N}^{-n0} + \sum_{k=1}^{N-1} Y(k) W_{2N}^{-nk} + \sum_{k=1}^{N-1} Y(2N-k) W_{2N}^{-n(2N-k)} + Y(N) \right]$$

Ahora vemos que es $W_{2N}^{-n(2N-k)}$

$$W_{2N}^{-n(2N-k)} = W_{2N}^{-2Nn} W_{2N}^{nk}$$

$$\begin{aligned}
 W_{2N}^{-2Nn} &= e^{-j \frac{2\pi}{2N} 2Nn} \\
 &= e^{-j\pi n} = 1
 \end{aligned}$$

entonces:

$$W_{2N}^{-n(2N-k)} = W_{2N}^{nk}$$

Con $Y(N) = 0$, tenemos que:

$$y(n) = \frac{1}{2N} \left[Y(0) + \sum_{k=1}^{N-1} Y(k) W_{2N}^{-nk} + \sum_{k=1}^{N-1} Y^*(k) W_{2N}^{nk} \right]$$

ya que

$$\begin{aligned}
 \left[Y(k) W_{2N}^{-nk} \right]^* &= Y^*(k) \left[W_{2N}^{-nk} \right]^* = Y^*(k) W_{2N}^{nk} \\
 &= Y(2N-k) W_{2N}^{nk}
 \end{aligned}$$

Las sumatorias tienen partes reales iguales, y partes imaginarias iguales pero de signo opuesto, entonces $Y(n)$ será igual a dos veces la parte real de una de las sumatorias, además $Y(0)$ es real, así:

$$y(n) = \frac{1}{2N} \operatorname{Re} \left[Y(0) + \sum_{k=1}^{N-1} Y(k) W_{2N}^{-nk} \right]$$

o de otra manera:

$$y(n) = \frac{1}{N} \operatorname{Re} \left[\frac{Y(0)}{2} + \sum_{k=1}^{N-1} Y(k) W_{2N}^{-nk} \right], \quad 0 \leq n \leq 2N-1 \quad (2.19)$$

Por sustitución de (2.13) en (2.19) tenemos:

$$y(n) = \frac{1}{N} \operatorname{Re} \left[\frac{C(0)}{2} + \sum_{k=1}^{N-1} C(k) W_{2N}^{(-n-1/2)k} \right];$$

$$0 \leq n \leq 2N-1$$

Utilizando (2.2) obtenemos:

$$f(n) = \frac{1}{N} \left[\frac{C(0)}{2} + \sum_{k=1}^{N-1} C(k) \cos \frac{(2n+1)k}{2N} \right], \quad 0 \leq n \leq N-1 \quad (2.20)$$

Las ecuaciones (2.12) y (2.20) forman un par de transformadas discretas coseno. Dado $C(k)$ para obtener $f(n)$, primero calculamos $Y(k)$ usando (2.13), luego usando (2.19) obtenemos la transformada de Fourier inversa compleja de $2N$ -puntos con lo cual resulta $y(n)$ y de allí $f(n)$.

2.4. TRANSFORMADA RAPIDA COSENO DIRECTA.

Lo que hacemos a continuación es obtener la transformada discreta coseno de la transformada discreta de Fourier de N -puntos de una secuencia real en vez de una transformada discreta de Fourier de $2N$ -puntos ahorrando la mitad de cálculos.

Primero dividimos la secuencia $y(n)$ en 2 secuencias de n -puntos.

$$\left. \begin{aligned} v(n) &= y(2n) \\ w(n) &= y(2n + 1) \end{aligned} \right\} 0 \leq n \leq N-1 \quad (2.21)$$

Donde $v(n)$ y $w(n)$ son los conjuntos de puntos par e impar respectivamente.

En la figura 2.3 podemos observar como tomamos estos puntos en el ejemplo.

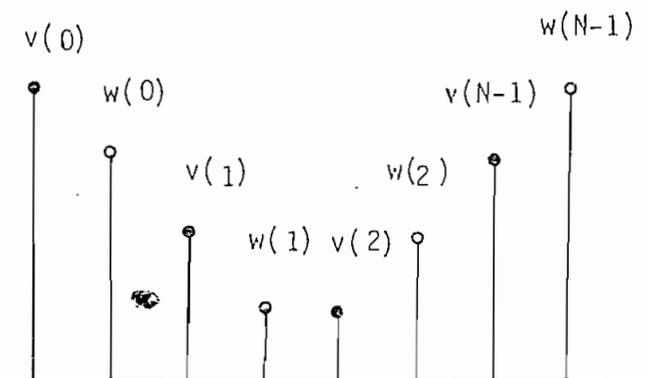


Fig. 2.3

Hay que notar que tanto $w(n)$ como $v(n)$ contienen todos los puntos de $f(n)$ y que $w(n)$ es la secuencia reversa de $v(n)$. Matemáticamente tomando (2.3) y (2.21).

$$w(n) = v(N-n-1), \quad 0 \leq n \leq N-1 \quad (2.22)$$

Sustituyendo (2.22) en (2.4)

$$Y(k) = \sum_{n=0}^{N-1} v(n) W_{2N}^{2nk} + \sum_{n=0}^{N-1} w(n) W_{2N}^{(2n+1)k} \quad (2.23)$$

Sustituyendo (2.22) en (2.23)

$$Y(k) = \sum_{n=0}^{N-1} v(n) W_N^{nk} + \sum_{n=0}^{N-1} v(N-n-1) W_{2N}^{(2n+1)k} \quad (2.24)$$

Haciendo el cambio de variables:

$$n' = N-n-1$$

$$n = N-n'-1$$

$$n = 0 \Rightarrow n' = N-1$$

$$n = N-1 \Rightarrow n' = 0$$

$$2n + 1 = 2(N-n'-1) + 1 = 2N - 2n' - 1$$

$$Y(k) = \sum_{n=0}^{N-1} v(n) W_N^{nk} + \sum_{n=0}^{N-1} v(n') W_{2N}^{2Nk - (2n'+1)k}$$

Haciendo: $n = n'$

$$: W \frac{2nk}{2N} = W \frac{4nk}{4N}$$

En definitiva pasamos todo a un mismo denominador $4N$, en el exponencial:

$$Y(k) = \sum_{n=0}^{N-1} v(n) W \frac{4nk}{4N} + \sum_{n=0}^{N-1} v(n) W \frac{-(4N+2)k}{4N}$$

Factorizando también es igual a:

$$Y(k) = W \frac{-k}{4N} \left[\sum_{n=0}^{N-1} v(n) W \frac{(4n+1)k}{4N} + \sum_{n=0}^{N-1} v(n) W \frac{-(4n+1)k}{4N} \right] \quad (2.25)$$

Los argumentos de las sumatorias son complejos conjugados, por lo tanto sus partes reales son iguales, por otra parte de (2.14) en (2.25) tenemos:

$$C(k) = W \frac{k/2}{2N} W \frac{-k}{4N} \left[\sum_{n=0}^{N-1} v(n) W \frac{(4n+1)k}{4N} + \sum_{n=0}^{N-1} v(n) W \frac{-(4n+1)k}{4N} \right], (2.26)$$

$$C(k) = 2 \operatorname{Re} \left[\sum_{n=0}^{N-1} v(n) W \frac{4nk}{4N} W \frac{k}{4N} \right] \quad (2.27)$$

Esto se puede expresar de dos maneras:

$$C(k) = 2 \operatorname{Re} \left[\sum_{n=0}^{N-1} \frac{W_N^{nk}}{4N} v(n) \right], \quad 0 \leq k \leq N-1 \quad (2.28)$$

O:

$$C(k) = 2 \sum_{n=0}^{N-1} v(n) \cos \frac{(4n+1)\pi k}{4N}, \quad 0 \leq n \leq N-1 \quad (2.29)$$

Esta última expresión es una definición alternativa de la transformada discreta coseno en términos de la secuencia reordenada $v(n)$ la cual puede ser escrita directamente en términos de $f(n)$.

$$v(n) = \begin{cases} f(2n), & 0 \leq n \leq \left\lceil \frac{N-1}{2} \right\rceil \\ f(2N-2n-1), & \left\lceil \frac{N+1}{2} \right\rceil \leq n \leq N-1 \end{cases} \quad (2.30)$$

Donde $\lceil a \rceil$ denota la parte entera de a .

En una inspección a esta última forma, vemos que $v(n)$ se obtiene tomando primero los puntos pares a $f(n)$ y luego los impares en orden reverso. Notar que (2.30), se aplica para N par o impar, también $v(n)$ es real, luego el algoritmo para calcular la transformada rápida coseno en el paso siguiente descrito en el capítulo 4, se realiza utilizando la secuencia compleja de $N/2$ - puntos en la transformada rápida de Fourier de $v(n)$.

2.5. TRANSFORMADA RAPIDA COSENO INVERSA

La idea aquí es calcular primero $v(n)$ a partir de la transformada discreta coseno, luego usando (2.30) obtenemos $f(n)$. Sustituyendo $v(n) = y(2n)$ en (2.19), y teniendo en cuenta que:

$$W_{2N}^{-2nk} = W_N^{-nk}$$

tenemos:

$$v(n) = \frac{1}{N} \operatorname{Re} \left[\frac{Y(0)}{2} + \sum_{k=1}^{N-1} Y(k) W_N^{-nk} \right], \quad 0 \leq n \leq N-1 \quad (2.31)$$

La ecuación anterior nos indica que $v(n)$ puede ser calculada usando una transformada discreta de Fourier inversa de N -puntos en vez de una de $2N$ -puntos que implica (2.19).

Hasta el momento tenemos cálculos alrededor de el doble que la transformada rápida discreta coseno directa.

Ahora demostraremos que la inversa, puede de hecho, ser calculada con el mismo número de cálculos que la transformada rápida coseno directa. El método es obtener $V(k)$ de $C(k)$, luego se calcula la transformada discreta de Fourier inversa de $V(k)$ para obtener $v(n)$.

La ecuación (2.28) puede ser reescrita como:

$$C(k) = \operatorname{Re} \left[2W_{4N}^k V(k) \right] \quad (2.32)$$

Donde $V(k)$ es la Transformada Discreta de Fourier de $v(n)$. Para calcular $V(k)$ de $C(k)$ en (2.32) necesitamos conocer también la parte imaginaria del término entre corchetes.

Para que no haya confusión, notemos todo el número complejo como $C_c(k)$ y la parte imaginaria como $C_i(k)$ donde:

$$C_c(k) = C(k) + j C_i(k) = 2W_{4N}^k V(k) \quad (2.33)$$

Entonces:

$$V(k) = \frac{1}{2} W_{4N}^{-k} C_c(k) \quad (2.34)$$

Ahora necesitamos determinar $C_i(k)$ para esto aprovechamos que $V(k)$ es simétrica hermiciana, es decir:

$$V(k) = V^{T*}(N-k) \quad (2.35)$$

donde:

$$V^{T*}(N-k) = [V_r(N-k) + j V_i(N-k)]^{T*} = V_i(N-k) - j V_r(N-k)$$

Notemos que (2.35) y (2.17) difieren en la definición de hermiciana, debido a que $V(k)$ es complejo, mientras que $Y(k)$ es real.

En este momento podemos sacar ventaja de reemplazar (2.37) en (2.13) para calcular $C(k)$, ya que podemos calcular simultáneamente $C(k)$ y $C(N-k)$.

De (2.37) tenemos:

$$V(k) = \frac{1}{2} W_{4N}^{-k} \left[C(k) - j C(N-k) \right], \quad 0 \leq k \leq N-1, \quad (2.38)$$

Calculamos $V(k)$ de (2.38) para $0 \leq k \leq N/2$, luego usamos (2.35) para $k \geq N/2$. Para calcular $V(0)$, únicamente necesitamos el valor de $C(N)$; este valor lo calculamos de (2.14) y (2.18).

$$C(N) = \sum_{n=0}^{N/2} Y(n)$$

De (2.18) tenemos:

$$Y(N) = 0$$

Entonces:

$$C(N) = 0 \quad (2.39)$$

Así, ya calculado $V(k)$, $v(n)$ se obtiene como la transformada rápida de Fourier inversa:

$$v(n) = \frac{1}{N} \sum_{k=0}^{N-1} V(k) W_N^{-nk} \quad (2.40)$$

Parecería de (2.40) que nuevamente tenemos que hacer tantos cálculos como en la transformada rápida de Fourier inversa de N-puntos.

Sin embargo en el capítulo 4, lo hacemos usando únicamente $N/2$ -puntos complejos con el mismo número de cálculos que la transformada rápida coseno directa.

3.1. TRANSFORMADA DE KARHUNEN LOEVE

En este capítulo se trata de dar una idea de lo que es la transformada de Karhunen Loéve, sin pensar que sea un estudio detallado sino más bien una orientación sobre este tópico, incluido en esta tesis, por tratarse de la transformada que optimiza los procesos de filtrado de Wiener desarrollados en el capítulo final.

Considérese un vector \mathbf{u} que proviene de un proceso (real) aleatorio gaussiano con valor medio cero y variancia R . Este vector se transforma linealmente mediante una matriz compleja T de $N \times N$ para producir un vector complejo \mathbf{v} cuyos componentes v_k no están mutuamente correlacionados. Cada componente v_k es cuantizada independientemente. El vector de salida \mathbf{v}' se transforma linealmente mediante una matriz T' para obtener un vector \mathbf{u}' . El problema es encontrar matriz óptima de decorrelación T y la matriz de reconstrucción T' , de tal manera que la distorsión media cuadrática promedio total:

$$D = \frac{1}{N} E \sum_{k=1}^N (u_k - u'_k)^2 \quad (3.1)$$

sea mínima. La solución a este problema se da cuando, [13] la matriz de reconstrucción óptima T' está dada por:

$$T' = T^{-1} P \quad (3.2)$$

Donde P es la matriz diagonal cuyos elementos son: [13]

$$\gamma_k = E (v_k v_k^*) / (E (v_k v_k^*)), \quad (3.3)$$

Donde E es el valor esperado y * denota el complejo conjugado.

La matriz de decorrelación óptima T es la Transformada Karhunen Loève de \mathbf{u} .

La Transformada Karhunen Loève tiene la propiedad, que para $M \leq N$ concentra el máximo de energía promedio en un número M de muestras de \mathbf{v} , $\{ v_i, i=1, \dots, N \}$.

Aunque la Transformada de Karhunen Loève es óptima, es difícil de calcular, no habiendo un algoritmo rápido asociado con ella.

Para secuencias aleatorias estacionarias, hay transformaciones unitarias de rápido cálculo que se aproximan a la eficiencia de la Transformada de Karhunen Loève.

Una de estas, es la transformada coseno la cual tiene una eficiencia muy próxima a la Transformada Karhunen Loève. Cuando el tamaño del vector tiende a infinito su eficiencia es equivalente a la Transformada de Karhunen Loève. Para determinados procesos la matriz de transformación coseno actúa en forma muy aproximada a la transformada óptima, cuando el parámetro de correlación se encuentra en el intervalo (0.5,1), aún cuando N sea pequeña.

Como ya hemos visto en capítulos anteriores la transformada coseno

puede calcularse mediante la transformada de Fourier, con un algoritmo rápido. Esta propiedad permite tener un sustituto útil de la Transformada de Karhunen Loève en algunos procesos donde las señales tienen alta correlación entre puntos.

Ahora, consideramos nuevamente el vector \mathbf{u} de $N \times 1$ componentes, si \mathbf{v} es su transformación entonces:

$$\mathbf{v} = T\mathbf{u} \quad (3-4)$$

Donde T es la matriz unitaria de $N \times N$ elementos, consideramos T como unitaria con el fin de asegurar la conservación de la energía de la señal en el dominio de la Transformada, es decir:

$$\sum_i |v_i|^2 = \sum_i |u_i|^2 \quad (3.5)$$

Existen varias transformadas unitarias como la de Fourier, Hadamard, Karhunen Loève, Slant, Haar, Coseno, etc. [16]. Las cuales concentran la mayor parte de energía en pocas muestras, las que son consideradas importantes en el transcurso del proceso. Generalmente estas muestras de interés se ubican en las bajas frecuencias.

Como hemos mencionado al iniciar este capítulo, nuestro interés no es hacer un desarrollo matemático profundo de la transformada de Karhunen Loève, sino por el contrario, dar una mínima información de orientación; de tal manera que prescindimos de los procesos deductivos y directamente formulamos la solución.

Esta matriz de transformación según A. K. Jain [17], para el proceso en el cual estamos interesados en aplicarla, como es El Filtrado de Wiener; se reduce a una matriz que tiene sus elementos dados por:

$$T_{ij} = \sqrt{\frac{2}{N+1}} \operatorname{Sen} \left(\frac{ij \pi}{N+1} \right) \quad (3.6)$$

Así tenemos una matriz real, con la cual haremos un filtrado subóptimo de Wiener en el capítulo final. Aunque no es la transformada ideal debido a que no hay un algoritmo de cálculo rápido asociado con ella, si es la que optimiza el filtrado.

4.1. ALGORITMO PARA EL CALCULO DE LA TRANSFORMADA RAPIDA COSENO

Ahora indicamos el procedimiento completo para calcular la Transformada Rápida Coseno, tanto directa como inversa.

El procedimiento para el cálculo de la Transformada Rápida Coseno Directa es el siguiente: [4].

Teniendo una secuencia real de N elementos, de la forma $f(n)$; $0 \leq n \leq N - 1$. Los pasos a seguir son:

1. De la ecuación (2.30) formamos la secuencia $v(n)$, así:

$$v(n) = \begin{cases} f(2n), & 0 \leq n \leq \left\lfloor \frac{N-1}{2} \right\rfloor \\ f(2N-2n-1), & \left\lceil \frac{N+1}{2} \right\rceil \leq n \leq N-1 \end{cases}$$

2. Con la Transformada Discreta de Fourier, calculamos $V(k)$, $0 \leq k \leq$

$$\left\lfloor \frac{N-1}{2} \right\rfloor \text{ a partir de } v(n).$$

3. Multiplicar $V(k)$ por $2 \exp(-j\pi k/2N)$

4. De (2.27):

$$C(k) - j C(N-k) = 2 W_{4N}^k V(k)$$

Simultáneamente calculamos $C(k)$ y $C(N-k)$. Notemos que, a pesar que k en $V(k)$ varía de $0 \leq k \leq \left\lceil \frac{N-1}{2} \right\rceil$; la variación de $C(k)$ es: $0 \leq k \leq N-1$.

El cálculo de la Transformada Rápida Coseno Inversa, dados $C(k)$, $0 \leq k \leq N-1$ y $C(N) = 0$, se logra: con el siguiente procedimiento:

1. Calculamos $V(k)$, como en (2.38).
2. Con la Transformada Discreta de Fourier Inversa, obtenemos $v(n)$.

Utilizando (2.30), recuperamos $f(n)$ de $v(n)$.

CONSIDERACIONES DE CALCULO

En la mayoría de computadores para aplicaciones científicas tienen desarrollado un algoritmo para el cálculo de la Transformada de Fourier, bien sea en su biblioteca de programas o directamente en su sistema operativo. En nuestra aplicación utilizamos una Transformada Rápida de Fourier, desarrollada para el computador Tektronix 4051 del que hacemos uso para las pruebas en la aplicación de Filtrado de Wiener, en el siguiente capítulo.

PROCEDIMIENTO PARA EL CALCULO DE LA TRANSFORMADA DIRECTA

Para ubicar los puntos pares e impares de $v(n)$ en las partes real e imaginaria respectivamente, nos ayudamos de un vector complejo.

$$t(n) = t_r(n) + t_i(n) \quad (4.1)$$

Definido así:

$$\left. \begin{array}{l} T_r(n) = v(2n) \\ t_i(n) = v(2n+1) \end{array} \right\} 0 \leq n \leq \left[\frac{N}{2} - 1 \right] \quad (4.2)$$

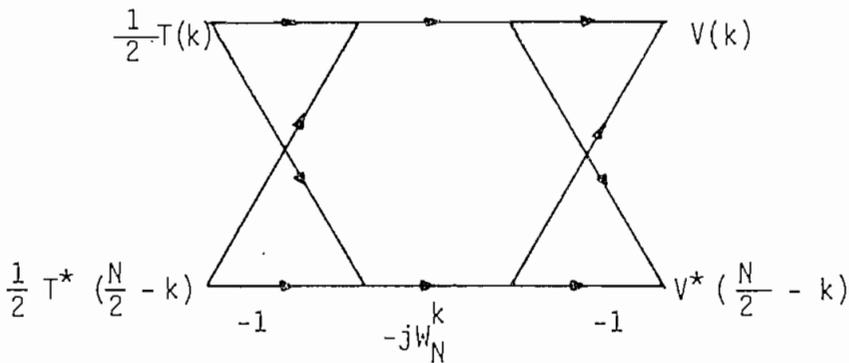


Fig. 4.1

Luego calculando la Transformada Discreta de Fourier de $\frac{N}{2}$ puntos de $t(n)$ y obtenemos $T(k)$ entre $0 \leq k \leq \frac{N}{2} - 1$.

Calcular $v(k)$ de $T(k)$ utilizando la fórmula dada en [5]

$$v(k) = \frac{1}{2} [T(k) + T^*(\frac{N}{2} - k)] - j \frac{1}{2} [T(k) - T^*(\frac{N}{2} - k)] \quad (4.3)$$

Los cálculos pueden hacerse más eficientes, de la siguiente manera a partir de (4.3) podemos escribir:

$$v^* \left(\frac{N}{2} - k \right) = \frac{1}{2} [T(k) + T^* \left(\frac{N}{2} - k \right)] + j \frac{1}{2} [T(k) - T^* \left(\frac{N}{2} - k \right)] \quad (4.4)$$

Dado $T(k)$, la figura 4.1, muestra el diagrama de flujo dado en [5] que implementa (4.3) y (4.4) para calcular $V(k)$.

Notar: que los valores de $V(k)$ son calculados dos al mismo tiempo, por consiguiente, el rango de los valores de k en la figura 4.1 será $0 \leq k \leq N/4$; los valores de $k \leq \frac{N}{2}$ podrán calcularse observando que $V(k)$ es hermiciano simétrico. Hay dos puntos en $V(k)$ que son reales y no requieren multiplicación alguna, ellos son:

$$V(0) = \text{Re} [T(0) + \text{Im} T(0)] \quad (4.5)$$

$$V \left(\frac{N}{2} \right) = \text{Re} [T(0) - \text{Im} T(0)] \quad (4.6)$$

Si N es divisible por cuatro tenemos [11]

$$V \left(\frac{N}{4} \right) = T^* \left(\frac{N}{4} \right) \quad (4.7)$$

PROCEDIMIENTO PARA EL CALCULO DE LA TRANSFORMADA INVERSA

Dado un $V(k)$ hermiciano simétrico, $0 \leq k \leq N-1$, deseamos calcular la Transformada Discreta de Fourier Inversa para obtener $v(n)$, $0 \leq n \leq N-1$.

De (4.3) y (4.4) podemos facilmente resolver para $T(k)$ y $T^*(N-k)$; el resultado de estas ecuaciones podemos implementarlo usando el diagrama de flujo de la figura 4.2 [5] donde $T(k)$ se calcula de $V(k)$ en el rango de $0 \leq k \leq \frac{N}{4}$

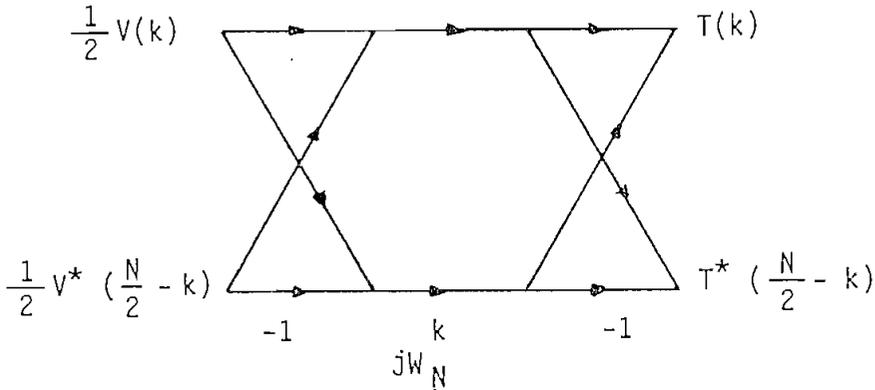


Fig. 4.2

Luego se calcula la Transformada Inversa de Fourier de $\frac{N}{2}$ puntos de $T(k)$, obteniéndose $t(n)$, $0 \leq n \leq \frac{N}{2} - 1$. Finalmente, usando (4.1) de $t(n)$ obtenemos $v(n)$.

En el caso de que N no sea divisible por 2 se puede calcular la Transformada Discreta de Fourier de dos secuencias separadas usando el mismo método anterior [12].

A continuación se muestra en detalle los algoritmos para la Transformada Rápida Coseno tanto directa como inversa.

```

000 REM TRANSPRID/COSENO  :*: TRANSFORMADA RAPIDA COSENO
010 F1=51
020 PRINT @P1:"JJJJ          TRANSFORMADA RAPIDA COSENO          JJJ"
030 PRINT @P1:"          NUMERO DE DATOSGGG"
040 INPUT M
050 N=M/2
060 PRINT @P1:"          ",M,"DATOS."
070 DELETE F,R,S,T,W,U,C,V
080 DIM F(M),R(N),S(N),T(2),U(2),W(2),C(M),V(M)
090 FOR I=1 TO M
100 PRINT @32:"GG"
110 INPUT F(I)
120 PRINT @P1: USING 1130:I,F(I)
130 IMAGE /,3X,"DATO F("3D" ) = ",5D,2D
140 NEXT I
150 REM :*: CALCULO DE LA TRANSFORMADA COSENO DIRECTA :*:
160 FOR I=1 TO N
170 J=I*2-1
180 V(I)=F(J)
190 NEXT I
200 K=N+1
210 FOR I=K TO M
220 K=I-1
230 L=2*(M-K)
240 V(I)=F(L)
250 NEXT I
260 FOR I=1 TO N
270 J=2*I
280 K=J-1
290 R(I)=V(K)
300 S(I)=V(J)
310 NEXT I
320 REM TRANSFORMADA COSENO UTILIZANDO FOURIER  :*:
330 F=PI
340 SET RADIANS
350 N2=N/2
360 N1=N-1
370 J=1
380 FOR I=1 TO N1
390 IF I=>J THEN 1460
400 T(1)=R(J)
410 T(2)=S(J)
420 R(J)=R(I)
430 S(J)=S(I)
440 R(I)=T(1)
450 S(I)=T(2)
460 K=N2
470 IF K=>J THEN 1510
480 J=J-K

```

(CONTINUACION)

```

490 K=K/2
500 GO TO 1470
510 J=J+K
520 NEXT I
530 M=1
540 L1=0
550 M=M*2
560 L1=L1+1
570 IF M<N THEN 1550
580 FOR L=1 TO L1
590 L2=2^L
600 L3=L2/2
610 U(1)=1
620 U(2)=0
630 W(1)=COS(F/L3)
640 W(2)=SIN(F/L3)
650 FOR J=1 TO L3
660 FOR I=J TO N STEP L2
670 G=I+L3
680 T(1)=R(G)*U(1)-S(G)*U(2)
690 T(2)=S(G)*U(1)+R(G)*U(2)
700 R(G)=R(I)-T(1)
710 S(G)=S(I)-T(2)
720 R(I)=R(I)+T(1)
730 S(I)=S(I)+T(2)
740 NEXT I
750 A1=U(1)*W(1)-U(2)*W(2)
760 A2=U(1)*W(2)+U(2)*W(1)
770 U(1)=A1
780 U(2)=A2
790 NEXT J
800 NEXT L
810 REM FIN DEL PROCESO DE FOURIER EN COSENO DIRECTA. **:
820 M=2*N
830 FOR K=1 TO N
840 L=K-1
850 CO=2*COS(F*L/M)
860 SO=2*SIN(F*L/M)
870 C(K)=R(K)*CO+SO*S(K)
880 J=M-L
890 C(J)=S(K)*CO-R(K)*SO
900 NEXT K
910 REM **: FIN TRANSFORMADA RAPIDA COSENO **:
920 FOR K=1 TO M
930 PRINT @P1: USING 1940:K,C(K)
940 IMAGE /,5X,"C ("3D" ) = ",5D,2D
950 NEXT K
960 END

```

TRANSFORMADA RAPIDA COSENO INVERSA

```

000 REM TRANSRAPII/COSENO  ;*: TRANSFORMADA RAPIDA COSENO INVERSA ;*:
010 P1=51
020 PRINT @P1:"JJJJ          TRANSFORMADA RAPIDA COSENO INVERSAJJJ"
030 PRINT @P1:"              NUMERO DE DATOSGGG"
040 INPUT M
050 N=M/2
060 PRINT @P1:"              ",M,"DATOS."
070 DELETE F,R,S,T,W,U,C,V
080 DIM F(M),R(N),S(N),T(2),U(2),W(2),C(M),V(M)
090 FOR I=1 TO M
100 PRINT @32:"GG"
110 INPUT C(I)
120 PRINT @P1: USING 1130:I,C(I)
130 IMAGE /,3X,"DATO  C('3D' ) = ",5D,2D
140 NEXT I
150 REM ;*: CALCULO DE LA TRANSFORMADA COSENO INVERSA ;*:
160 P=PI
170 SET RADIANS
180 FOR K=1 TO N
190 CO=-COS(P*(K-1)/M)
200 SO=SIN(P*(K-1)/M)
210 S(K)=C(K)*SO-C(M-K+1)*CO.
220 R(K)=C(K)*CO+C(M-K+1)*SO
230 NEXT K
240 REM TRANSFORMADA COSENO INVERSA UTILIZANDO FOURIER INVERSA ;*:
250 N2=N/2
260 N1=N-1
270 J=1
280 FOR I=1 TO N1
290 IF I=>J THEN 1360
300 T(1)=R(J)
310 T(2)=S(J)
320 R(J)=R(I)
330 S(J)=S(I)
340 R(I)=T(1)
350 S(I)=T(2)
360 K=N2
370 IF K=>J THEN 1410
380 J=J-K
390 K=K/2
400 GO TO 1370
410 J=J+K
420 NEXT I
430 M=1
440 L1=0
450 M=M*2
460 L1=L1+1

```

(CONTINUACION)

```

470 IF M<N THEN 1450
480 FOR L=1 TO L1
490 L2=2^L
500 L3=L2/2
510 U(1)=1
520 U(2)=0
530 W(1)=COS(P/L3)
540 W(2)=SIN(P/L3)
550 FOR J=1 TO L3
560 FOR I=J TO N STEP L2
570 G=I+L3
580 T(1)=R(G)*U(1)-S(G)*U(2)
590 T(2)=S(G)*U(1)+R(G)*U(2)
600 R(G)=R(I)-T(1)
610 S(G)=S(I)-T(2)
620 R(I)=R(I)+T(1)
630 S(I)=S(I)+T(2)
640 NEXT I
650 A1=U(1)*W(1)-U(2)*W(2)
660 A2=U(1)*W(2)+U(2)*W(1)
670 U(1)=A1
680 U(2)=A2
690 NEXT J
700 NEXT L
710 REM FIN DEL PROCESO DE FOURIER INVERSO EN COSENO INVERSA, :*:
720 M=2*N
730 R=-R
740 R=R/M
750 S=S/M
760 J=0
770 FOR K=1 TO N2
780 L=N-K+1
790 J=J+1
800 F(J)=R(K)
810 J=J+1
820 F(J)=S(L)
830 J=J+1
840 F(J)=S(K)
850 J=J+1
860 F(J)=R(L)
870 NEXT K
880 FOR K=1 TO M
890 PRINT @P1: USING 1900:K,F(K)
900 IMAGE /,5X,"*- F(",3D," ) = "5D,2D
910 NEXT K
920 END

```

4.2. ALGORITMO PARA EL CALCULO DE LA TRANSFORMADA DE KARHUNEN LOEVE

El programa que hemos elaborado, al correrlo, pide primero el número de datos y enseguida los datos; hay una restricción en el número de datos ya que tiene que ser potencia de dos.

Una vez ingresado los datos, el programa genera la matriz de transformación de Karhunen Loéve utilizando la ec. (3.6):

$$T_{ij} = \sqrt{\frac{2}{N+1}} \operatorname{sen} \left(\frac{ij\pi}{N+1} \right) \quad (4.8)$$

Los resultados se imprimen con sus respectivas leyendas al finalizar los cálculos de transformación entre los datos y la matriz generada con (4.8).

Para la Transformada de Karhunen Loéve Inversa, utilizamos la misma directa como inversa, ya que las matrices generadas por la fórmula (4.8) tienen la particularidad de ser ellas mismo su matriz inversa.

El programa que utilizamos para el Filtrado subóptimo de Wiener, lo listamos a continuación:

```

1000 REM      *KARHUNEN/LOEVE*
1030 REM
1040 REM      PROGRAMA PARA HACER
1050 REM      LA TRANSFORMADA KARHUNEN LOEVE
1060 REM      E, P, N.
1070 REM      JAIME ARTURO ERAZO
1100 PRINT *LJINGRESE LOS DATOS*
1110 REM LOS DATOS SE INGRESAN SECUENCIALMENTE
1120 PRINT *JEL ORDEN DE LOS DATOS ES: *;
1130 INPUT N
1140 DELETE F
1150 DIM F(N)
1160 FOR I=1 TO N STEP .4
1170 PRINT *JDATO(";I;")=";
1180 INPUT F(I)
1190 J=I+1
1200 PRINT *K          DATO(";J;")=";
1210 INPUT F(J)
1220 J=J+1
1230 PRINT *K          DATO(";J;")=";
1240 INPUT F(J)
1250 J=J+1
1260 PRINT *K          DATO(";J;")="
1270 INPUT F(J)
1280 NEXT I
1400 P1=51
1410 PRINT @P1:"LJJJJ          TRANSFORMADA"
1420 PRINT @P1:"J          K A R H U N E N   L O E V E"
1430 PRINT @P1:"=====
1440 PRINT @P1:"JJEL NUMERO DE DATOS ES : ";N
1450 PRINT @P1:"JJLOS DATOS SON:JJ",F
1490 REM AQUI: TRANSFORMADA KARHUNEN LOEVE          * INICIO *
1500 DELETE A,F1,C
1510 DIM A(N,N),F1(N,N),C(N)
1530 P=PI
1540 SET RADIANS
1550 N1=N+1
1560 FOR I=1 TO N
1570 K=I+1
1580 F1(I,I)=SQR(2/N1)*SIN(I*I*P/N1)
1590 IF K>N THEN 1650
1600 FOR J=K TO N
1610 F1(I,J)=SQR(2/N1)*SIN(I*J*P/N1)
1620 F1(J,I)=F1(I,J)
1630 NEXT J
1640 NEXT I
1650 REM FIN DE LOS CALCULOS DE LA MATRIZ K.L.
1660 FOR J=1 TO N
1670 C(J)=0
1680 FOR K=1 TO N
1690 C(J)=C(J)+F(K)*F1(J,K)
1700 NEXT K
1710 NEXT J
1720 REM SE TRANSFORMO EL VECTOR DE ENTRADA POR KARHUNEN LOEVE (FIN)
1730 REM
1740 PRINT @51:"JJY LOS RESULTADOS SON : ",C
1750 END

```

FILTRADO DE WIENER

El filtrado de Wiener es una técnica clásica de tratar señales, en un principio se aplicó a señales continuas en el tiempo y unidimensionales, los análisis y la implementación se basaron en la teoría de Fourier para señales continuas [6]. Por supuesto, es posible aplicar el filtrado de Wiener en señales muestreadas en el tiempo, a demás extender la técnica a dos dimensiones.

El filtrado puede implementarse con transformaciones unitarias como: la de Fourier, Coseno, entre otros [7], [9].

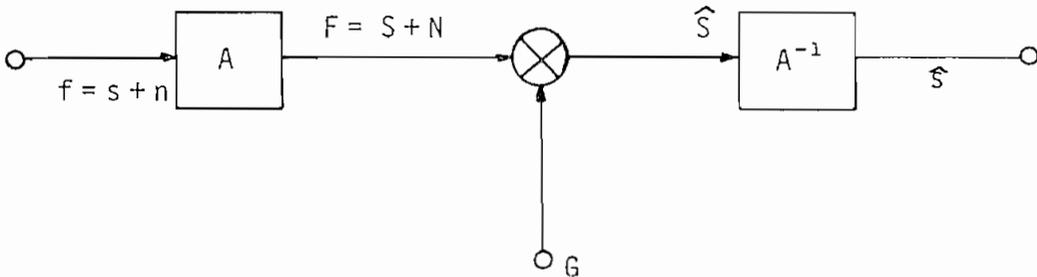


Fig. 5.1

Filtro General de Wiener

La figura 5.1, representa el diagrama en bloques de un sistema generalizado de filtrado de Wiener en una dimensión.

El vector f es un vector columna con M elementos de valor promedio cero; formado por la suma de la señal s más el ruido n , las que forman la entrada del sistema. Se asume que la señal y el ruido no están correlacionados entre sí.

En el sistema la matriz A de $M \times M$ elementos, define una operación de transformación unitaria. La Transformación nos entrega el vector:

$$F = Af = As + An = S + N \quad (5.1)$$

Este vector F es multiplicado por una matriz filtro G de orden $M \times M$. Hay que notar que no siempre G es una matriz diagonal. En el caso que contenga términos fuera de la diagonal, el filtrado es llamado Filtrado Vectorial, este combina todas las componentes espectrales del vector de entrada o dato. Si por otro lado G no contiene términos fuera de la diagonal, el filtrado es llamado simplemente Escalar, el cual toma individualmente cada componente del espectro del vector de entrada.

Finalmente, con la operación inversa de la transformación unitaria estimamos el resultado,

$$\hat{s} = A^{-1} G F \quad (5.2)$$

$$\hat{s} = A^{-1} G A f \quad (5.3)$$

La matriz filtro G conjuntamente con la matriz de transformación son de la clase que minimizan el error medio cuadrático entre la señal y su valor esperado.

La meta en el diseño de los filtros, es por supuesto, mantener el error dentro de niveles aceptables, inclusive llegando a mínimo ab

soluto cuando se trata del filtro óptimo. Eventualmente al tratar de simplificar los requerimientos computacionales se diseña un filtro subóptimo, seleccionando ciertos elementos de G para hacerlos ceros, con el fin de obtener un algoritmo rápido en el desarrollo de la matriz filtro.

Habiendo la posibilidad de utilizar una transformada para el filtro, hay que calcular la matriz G . En general la matriz filtro tiene componentes [9]

$$g(i,j) = \sum_{t=1}^N K_S(t,i) K_S(t,j) \frac{\lambda_S(t)}{k_0 + \lambda_S(t)}, \quad (5.4)$$

donde K_S y λ_S son los vectores propios y valores propios, respectivamente, de la matriz de covariancia:

$$C_S = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & 1 & \rho & \dots & \rho^{N-2} \\ \rho^2 & \rho & 1 & \dots & \rho^{N-3} \\ \dots & \dots & \dots & \dots & \dots \\ \rho^{N-1} & \rho^{N-2} & \rho^{N-3} & \dots & 1 \end{bmatrix} \quad (5.5)$$

Donde ρ es la correlación entre elementos adyacentes. Con el ánimo de aliviar los requerimientos computacionales, optamos por hacer el Filtrado Subóptimo de Wiener en vez del Filtrado Óptimo de Wiener.

FILTRADO DE WIENER SUBOPTIMO

En el presente capítulo hacemos uso de este tipo de filtrado, por la facilidad de implementarlo, memoria disponible y rapidez de ejecución del computador Tektronix 4051.

Una buena alternativa para generar la matriz de filtro subóptimo, es la que da Pratt [9]; mediante un proceso corto, donde se obtiene un filtro escalar relativamente sencillo, dado por

$$G = \text{diag.} \left[\frac{C_S(i,i)}{C_S(i,i) + C_N(i,i)} \right] \quad (5.6)$$

Donde diag. nos representa los términos de la diagonal, los elementos restantes de G son ceros; y

$$C_S = A C_s A^{*T} \quad (5.7)$$

$$C_N = A C_n A^{*T} \quad (5.8)$$

Siendo A una matriz de transformación unitaria como las ya mencionadas (Fourier, Coseno, Karhunen Loève, etc.) C_s está definida por (5.5) y C_n la matriz de correlación del ruido dada por:

$$C_n = \begin{bmatrix} k_0 & & & & \\ & k_0 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & k_0 \end{bmatrix} \quad (5.9)$$

Aquí k_0 es la relación directa entre las densidades de potencia del ruido y la señal.

Para realizar los filtrados tomaremos una señal, senoidal la que muestreemos con $N = 16$. Sería interesante aumentar N ya que el filtrado mejora notablemente cuando la cantidad de muestras es mayor, pero la limitación por la capacidad de memoria disponible en el sistema utilizado no permite sobrepasar las 16 muestras.

Después de muestrear la señal introducimos ruido aleatorio directamente a cada muestra, este ruido es generado internamente por el computador.

A continuación mostramos los pasos seguidos en el filtrado y sus resultados.

5.1 SOLUCION POR LA TRANSFORMADA COSENO

El Filtrado Subóptimo de Wiener mediante la Transformada Rápida Coseno, se realiza con programas que se cargan en la memoria del computador TEKTRONIX 4051 comandados por el programa "MAESTRO".

El Diagrama de Bloques del Filtrado de Wiener utilizando la Transformada Rápida Coseno se indica en la figura 5.2.

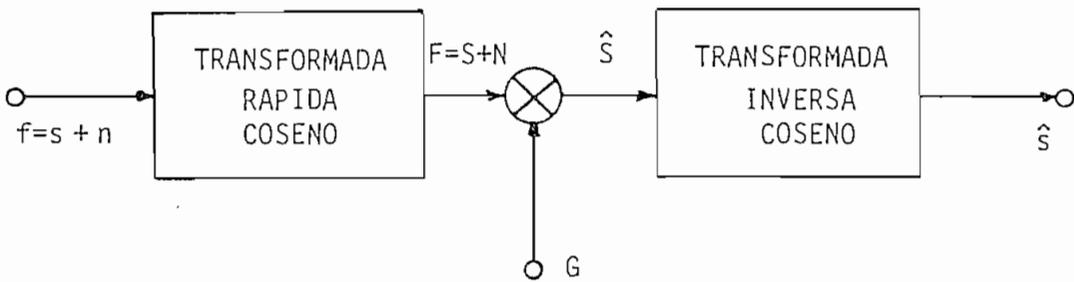


Fig. 5.2

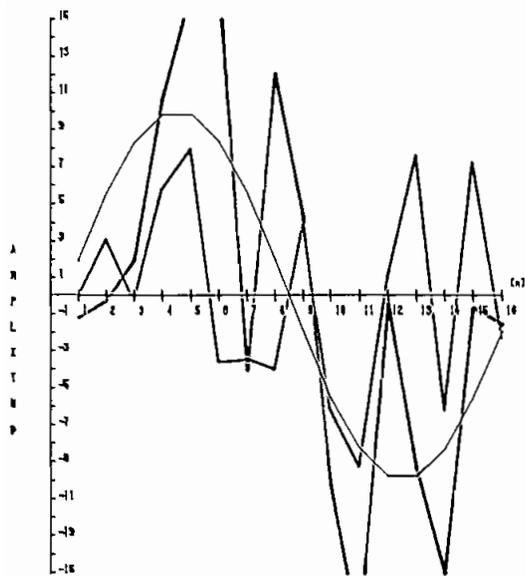
Filtrado de Wiener por Transformada Rápida Coseno

En la figura anterior G representa una matriz diagonal con elementos dados en las ecuaciones (5.6), (5.7), (5.8) y (5.9); en las que la matriz A ha sido reemplazada por la Transformada Rápida Coseno; f representa la señal s más el ruido n , y \hat{s} es la estimación de la señal.

La figura 5.3 representa: La señal sin ruido (en color negro); la señal con ruido (en color rojo), y la estimación de la señal resultado del Filtrado (en color verde).

FILTRADO POR TRANSFORMADA

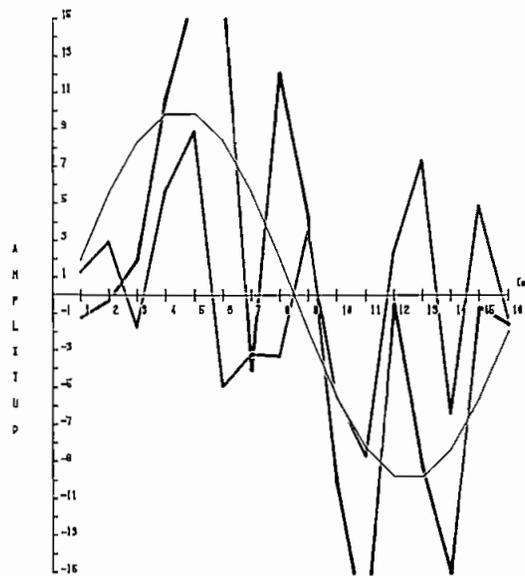
COSENO



FACTOR DE CORRELACION - 0.5
RELACION BILATERAL - 0.56

FILTRADO POR TRANSFORMADA

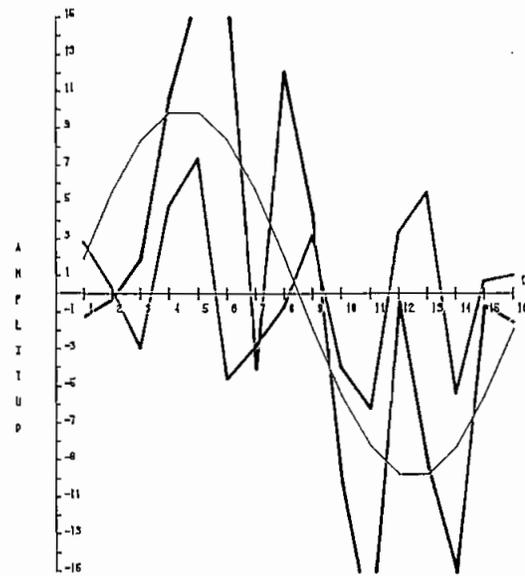
COSENO



FACTOR DE CORRELACION - 0.6
RELACION BILATERAL - 0.56

FILTRADO POR TRANSFORMADA

COSENO



FACTOR DE CORRELACION - 0.56
RELACION BILATERAL - 0.56

Fig. 5.3

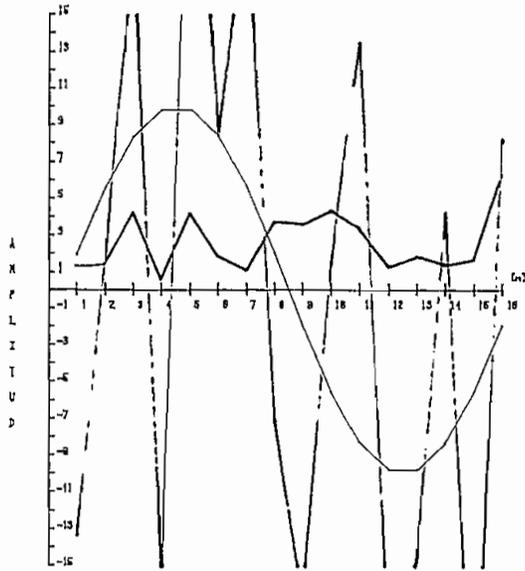
5.2 OTRAS SOLUCIONES

Cambiando la Transformada Rápida Coseno por las Transformadas de Fourier y Karhunen Loéve se obtienen otras soluciones en el Filtrado Subóptimo de Wiener. Para el efecto se sustituyen la matriz A por las transformadas mencionadas en las ecuaciones (5.3), (5.7) y (5.8) de la misma forma como se hizo en el numeral 5.1.

Los gráficos tienen la convención de colores del numeral anterior para las figuras 5.4 y 5.5.

FILTRADO POR TRANSFORMADA

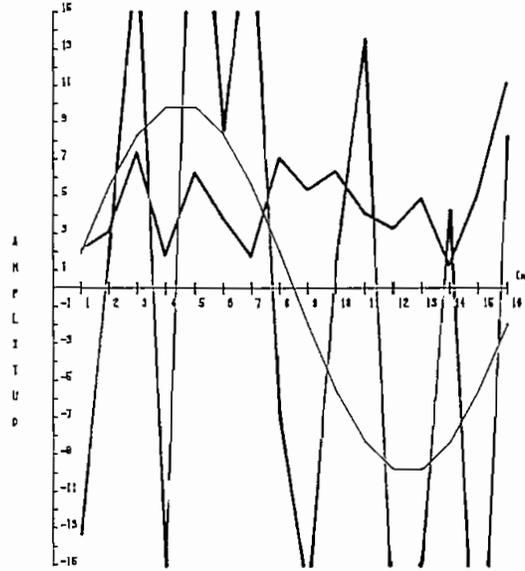
DE FOURIER



FACTOR DE CORRELACION - 0.5
RELACION RUIDO/SEÑAL - 2

FILTRADO POR TRANSFORMADA

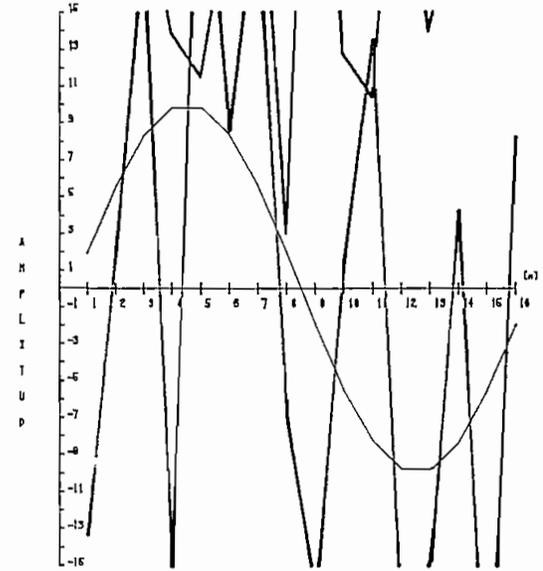
DE FOURIER



FACTOR DE CORRELACION - 0.7
RELACION RUIDO/SEÑAL - 2

FILTRADO POR TRANSFORMADA

DE FOURIER

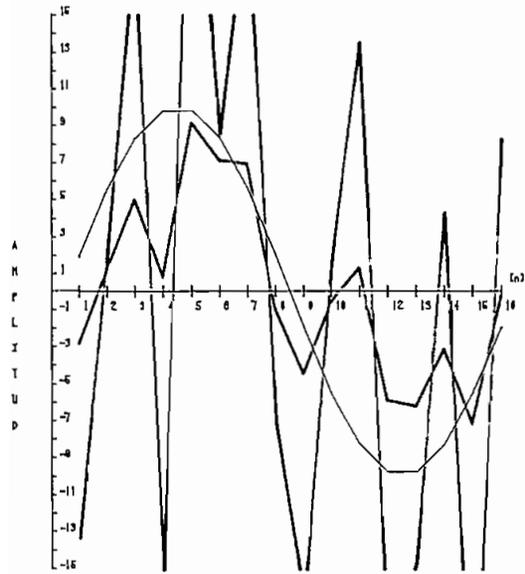


FACTOR DE CORRELACION - 0.95
RELACION RUIDO/SEÑAL - 2

Fig. 5.4

FILTRADO POR TRANSFORMADA

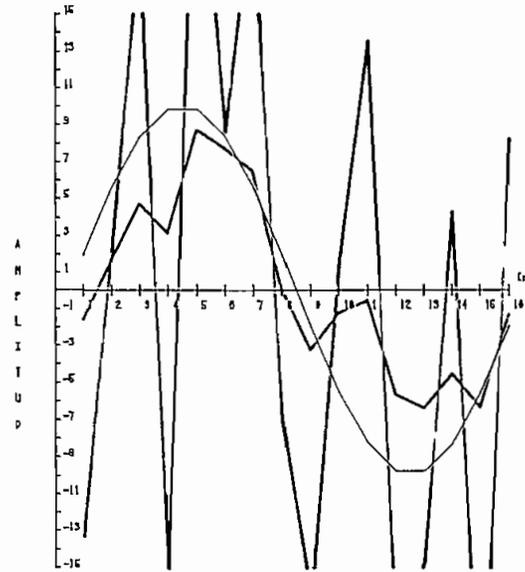
KARHUNEN LOEVE



FACTOR DE CORRELACION - 0,6
RELACION RUIDO/SEÑAL - 2

FILTRADO POR TRANSFORMADA

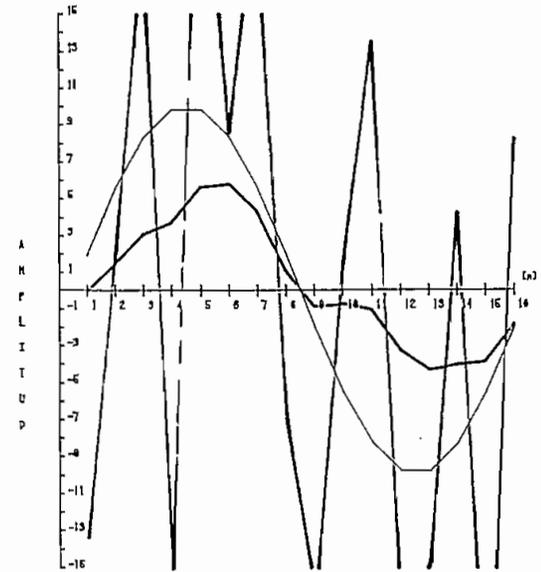
KARHUNEN LOEVE



FACTOR DE CORRELACION - 0,7
RELACION RUIDO/SEÑAL - 2

FILTRADO POR TRANSFORMADA

KARHUNEN LOEVE



FACTOR DE CORRELACION - 0,96
RELACION RUIDO/SEÑAL - 2

Fig. 5.5

5.3 COMPARACION DE RESULTADOS

En la figura 5.6 se muestra la señal senoidal sin ruido (en color negro) y la señal con ruido (en color rojo); su relación ruido/señal es 1.3.

Con el factor de correlación de 0.9 se ha realizado el proceso de Filtrado Subóptimo de Wiener por las Transformadas: Rápida Coseno, Rápida de Fourier y de Karhunen Loève. En la figura 5.6 se ha graficado la estimación de la señal (en color verde) obtenida después del proceso de Filtrado por las Transformadas mencionadas.

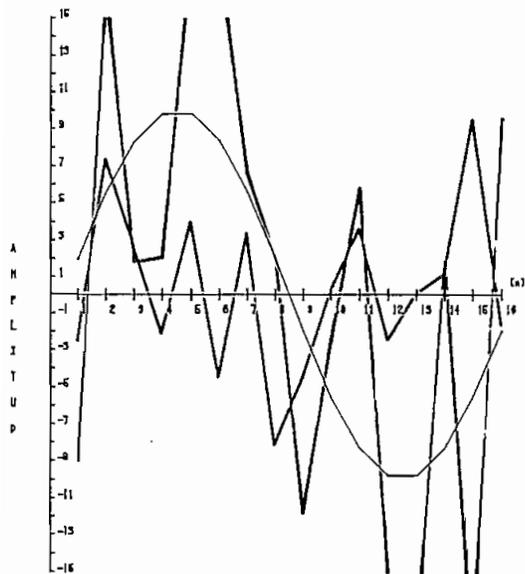
Notemos que la mejor estimación de la señal es la que se hace mediante la Transformada de Karhunen Loève. Le sigue en calidad la estimación realizada por la Transformada Rápida Coseno. La estimación con el Filtrado de Wiener vía la Transformada Rápida de Fourier, no es buena, pues sobrepasa los límites del gráfico más que la señal con ruido.

En la figura 5.7 se puede observar con la convención de colores indicada anteriormente, una misma señal de entrada con diferentes relaciones de ruido/señal (2.1 y 0.33), es notorio que la estimación mejora en la medida que el ruido disminuye.

Tomando en cuenta que la señal filtrada es una función Seno y que cualquier señal puede descomponerse en la suma de funciones senoidales podemos concluir que las observaciones realizadas son aplicables en forma general a una señal periódica muestreada N veces, don

FILTRADO POR TRANSFORMADA

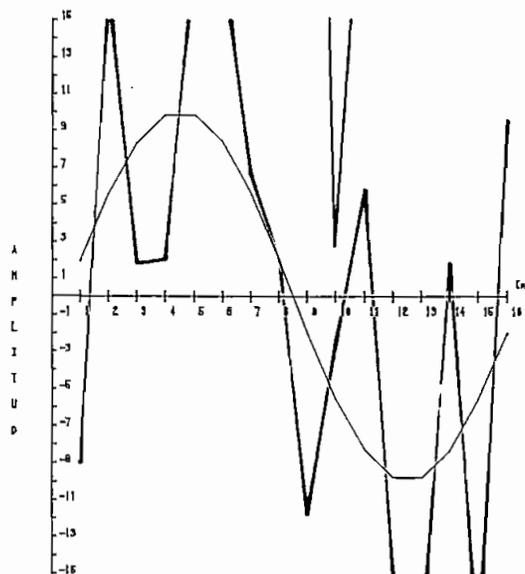
COSENO



FACTOR DE CORRELACION - 0.9
 RELACION RUIDO/SEÑAL - 1.5

FILTRADO POR TRANSFORMADA

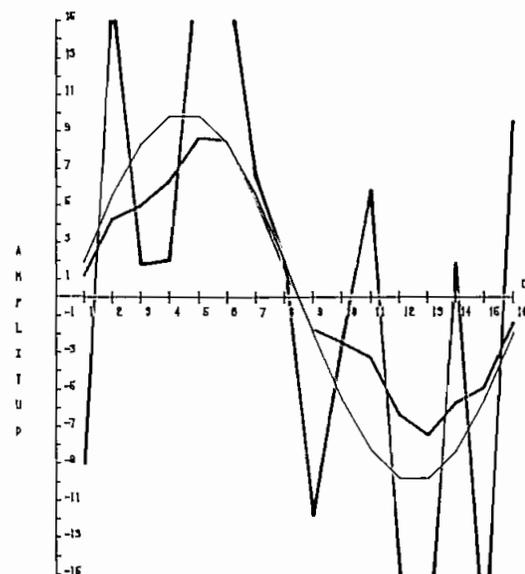
DE FOURIER



FACTOR DE CORRELACION - 0.9
 RELACION RUIDO/SEÑAL - 1.5

FILTRADO POR TRANSFORMADA

KARHUNEN LOEVE

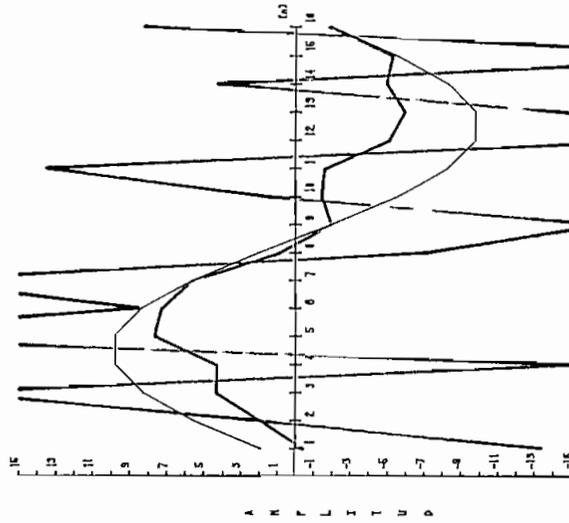


FACTOR DE CORRELACION - 0.9
 RELACION RUIDO/SEÑAL - 1.5

Fig. 5.6

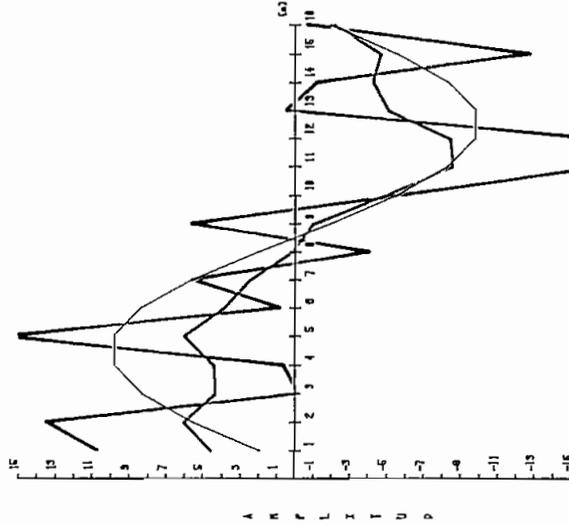
FILTRADO POR TRANSFORMADA

KARHUNEN LOEVE



FILTRADO POR TRANSFORMADA

KARHUNEN LOEVE



FILTRADO POR TRANSFORMADA

KARHUNEN LOEVE

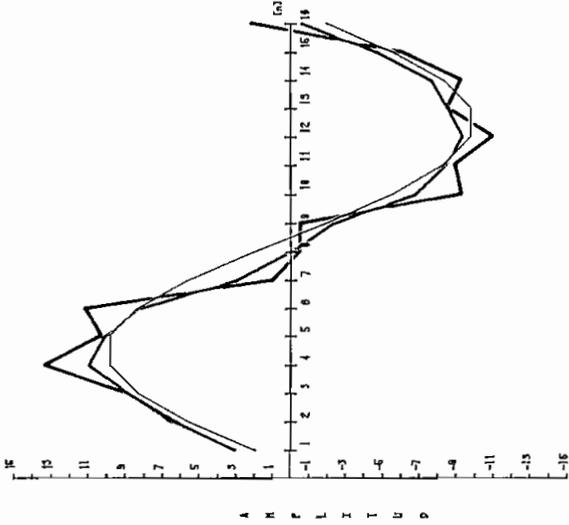
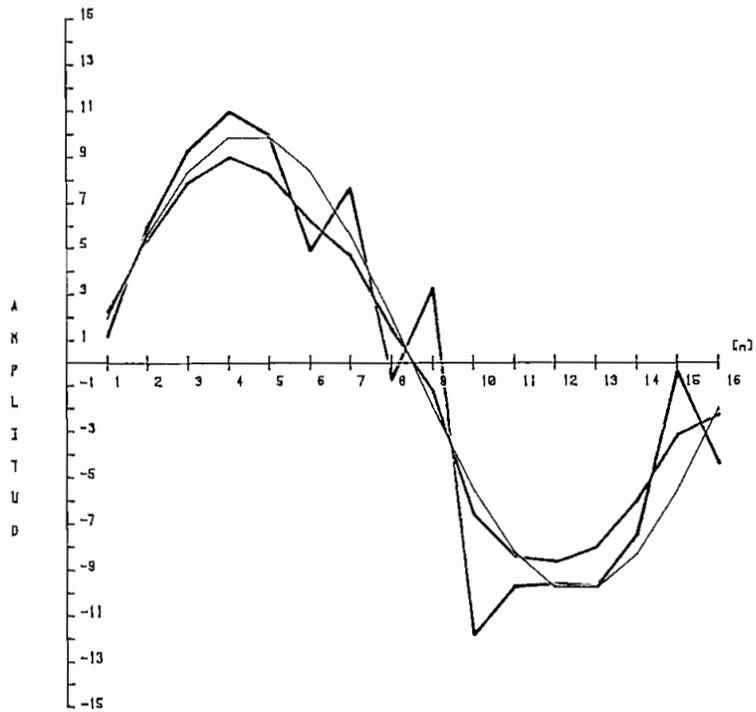


Fig. 5.7

En la figura 5.8 se observa a la izquierda el proceso de filtrado cuando la señal ha sido alterada por un ruido Gaussiano y a la derecha sin variar los parámetros con la misma señal pero con una alteración de un ruido uniformemente distribuido. Se nota que los resultados de filtrado son mejores cuando la señal se altera con ruido Gaussiano, esto resulta interesante debido a que los procesos donde se genera ruido en la naturaleza poseen una distribución de tipo Gaussiano y es precisamente donde el filtrado posee mejores características.

FILTRADO POR TRANSFORMADA

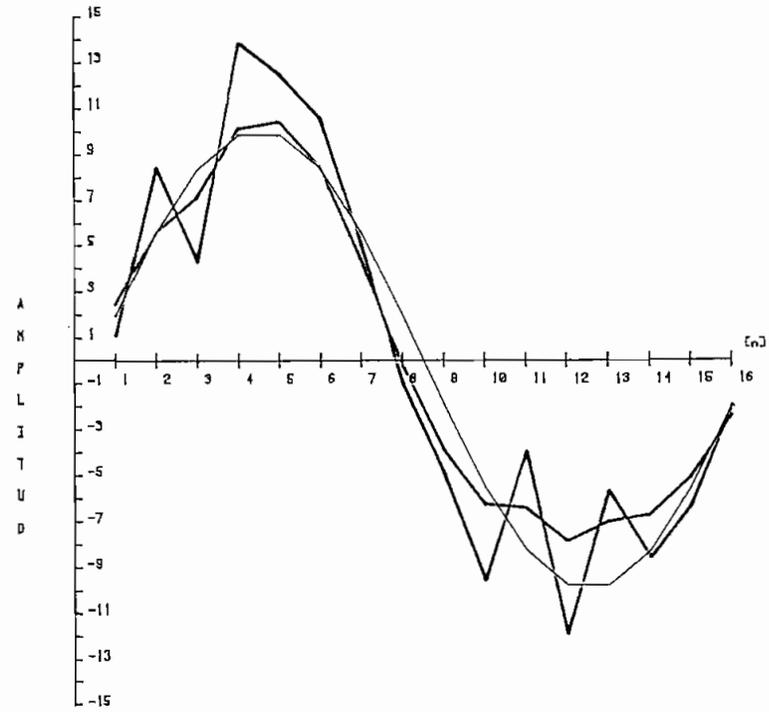
KARHUNEN LDEVE



FACTOR DE CORRELACION - 8.9
 RELACION RUIDO/SEÑAL - 8.4

FILTRADO POR TRANSFORMADA

KARHUNEN LOEVE



FACTOR DE CORRELACION - 8.9
 RELACION RUIDO/SEÑAL - 8.4

Fig. 5.8

5.4 CONCLUSIONES

Después de observar los rendimientos computacionales en el Filtrado de Wiener por las diferentes Transformadas, con especial atención en la Transformada Rápida Coseno se puede concluir:

A.- Que el Filtrado de Wiener se puede realizar mediante el uso de Transformadas Unitarias.

B.- La capacidad de memoria a utilizar en el computador, depende de la transformada elegida. La cantidad de memoria requerida para las instrucciones de nuestros programas es la siguiente:

TRANSFORMADA RAPIDA COSENO 6415 BYTES

TRANSFORMADA RAPIDA DE FOURIER 5210 BYTES

TRANSFORMADA DE KARHUNEN LOEVE 2792 BYTES

Observemos que la Transformada Rápida Coseno es la que más memoria utiliza, debido a que además de contener las instrucciones de la Transformada Rápida de Fourier, posee instrucciones para ubicar específicamente los datos y recuperar los resultados en forma ordenada.

La Transformada de Karhunen Loéve ocupa menos memoria ya que se implementa directamente la fórmula, en BASIC y no como un algoritmo para ahorrar tiempo de cálculo como las anteriores.

C.- La rapidez de cálculo con que se obtienen los resultados, depende directamente del algoritmo disponible para la respectiva

Transformada. En nuestros programas se obtienen los siguientes tiempos cuando el número de muestras es $N = 16$:

TRANSFORMADA RAPIDA COSENO 13 SEGUNDOS

TRANSFORMADA RAPIDA DE FOURIER 34 SEGUNDOS

TRANSFORMADA KARHUNEN LOEVE 47 SEGUNDOS

La Transformada Rápida Coseno es la más rápida de las tres, y podría pensarse que su cálculo requiera la mitad del tiempo utilizado por la Transformada Rápida de Fourier, ya que en la primera los datos son la mitad de los de la segunda en un mismo algoritmo. Sin embargo, difiere debido precisamente a que antes y después de utilizar el algoritmo de la Transformada Rápida de Fourier, se organiza los datos adecuadamente.

La Transformada Karhunen Loéve en este aspecto es lenta respecto a las demás ya que no se posee un algoritmo rápido asociado con ella, sino que directamente se implementa la fórmula con la versatilidad que da en estos casos el lenguaje BASIC.

Nótese bien que los tiempos registrados para cada transformada son los que se obtienen cuando en el archivo del filtro se lee el primer dato, en consecuencia, estos tiempos no son constantes varían de acuerdo a la dimensión de los datos y la ubicación de la matriz filtro dentro del archivo, aún más, si no está disponible esta matriz el programa se encarga de indicar esta situación y automáticamente se actualiza el archivo con la consiguiente demora extra.

MANUAL DE OPERACION

Lo que a continuación se describe, es la manera como se corren los programas al realizar un Filtrado de Wiener.

El momento de encender el equipo hay que hacerlo siguiendo un orden establecido. Este orden debe llevarse estrictamente para un correcto manejo del computador. Inicialmente se deben encender las unidades de diskette 1, 2 y a continuación la 0, luego proceder a encender el computador TEKTRONIX 4051; una vez realizado esto, se carga la fecha y la hora antes de trabajar con los diskettes, en la siguiente forma:

```
CALL "SETTIME", "DD-MMM-AA HH:MM:SS"
```

Junto al llamado de la subrutina "SETTIME" se debe poner entre comillas, la fecha del día, un guión, las tres primeras iniciales del mes en inglés, un guión, las dos últimas cifras del año, un espacio en blanco, las horas, dos puntos, los minutos y opcionalmente dos puntos y los segundos, antes de cerrar las comillas.

Una vez ingresado el diskette en su controlador, hay que hacer saber al computador que este está disponible y realizar el montaje; para esto se ejecuta la siguiente subrutina.

```
CALL "MOUNT",D9,A$
```

D9 se refiere a la unidad donde se aloja el diskette y A\$ es una

variable donde se graba la identificación del diskette, especificada durante el proceso de formateo.

Para cargar el programa "MAESTRO" en la memoria RAM del computador se ejecuta la siguiente instrucción:

```
OLD "MAESTRO"
```

El programa puede empezar a correr ingresando un RUN por el teclado o simplemente eligiendo la tecla # 1, con lo que aparecerá en pantalla el índice de programas accesibles al usuario a través del programa "MAESTRO". Estas opciones de programas son:

TECLA 1 INDICE DE PROGRAMAS

TECLA 2 INGRESO DE DATOS

TECLA 3 FILTRADO POR LA TRANSFORMADA RAPIDA COSENO

TECLA 4 FILTRADO POR LA TRANSFORMADA RAPIDA DE FOURIER

TECLA 5 FILTRADO POR LA TRANSFORMADA DE KARHUNEN LOEVE

TECLA 6 GRAFICOS DE DATOS Y RESULTADOS

FORMA DE FUNCIONAMIENTO

Una vez cargado el programa "MAESTRO" se habilitan las teclas predefinidas para el usuario mediante las cuales se realiza la carga automática de los diferentes programas que se describen a continuación.



INGRESO DE DATOS

La tecla 2- carga el programa "INGRESO/DATOS" desde la instrucción número 1000 y lo ejecuta. Solamente en la primera ejecución preguntará por la unidad donde está el disco, cuyo valor puede ser 0, 1 ó 2. A continuación se debe ingresar la longitud del vector de datos e inmediatamente los datos, el factor de correlación y la relación ruido/señal. Si se desea incluir ruido en la señal se responderá con "S" o "SI" a la pregunta formulada por el programa. En caso afirmativo, si el ruido deseado es gaussiano se ingresa la cantidad de números aleatorios uniformemente distribuidos a sumarse; en el caso que se quiera ruido uniformemente distribuido se ingresa un 0. Luego de los cálculos del ruido, este aparecerá en pantalla.

EJECUCION DE LOS PROGRAMAS

Antes de elegir el filtrado por uno de los tipos de transformadas disponibles es indispensable haber ingresado los datos.

Si se desea la impresión de datos y resultado durante la ejecución del filtrado se alistará el impresor antes de aplastar cualquier tecla predefinida para el usuario. Las teclas predefinidas para Filtrado de Wiener son las siguientes:

TECLA 3 Ejecutará un filtrado por la Transformada Rápida Coseno.

TECLA 4 Ejecutará un filtrado por la Transformada Rápida de Fourier.

TECLA 5 Ejecutará un filtrado por la Transformada de Karhunen Loéve.

Eventualmente, si en los archivos de filtro no se dispone de la ma
triz filtro, se procede automáticamente a actualizarlo y se retor
na al programa elegido.

GRAFICOS DE DATOS Y RESULTADOS

Al graficar los datos y resultados se debe alistar el graficador y colocar la pluma con tinta negra. Al aplastar la tecla 6 se dibu
ja los ejes y la señal sin el ruido, a continuación viene una inte
rrupción para cambiar de pluma a una con tinta roja, la que traza la señal con ruido después de un RETURN. Luego de la siguiente in
terrupción donde se requiere cambiar a tinta verde para que se tra
ce la estimación de la señal resultado del filtrado. Al final se coloca la pluma inicial para terminar los detalles del gráfico (tí
tulos, escalas, datos numéricos y nombres).


```

00 REM PROGRAMA "INGRESO/DATOS", PARA INGRESAR LOS DATOS
10 Q2=2
20 GO TO 800
30 PRINT "LJINGRESE LOS DATOS"
40 REM LOS DATOS SE INGRESAN SECUENCIALMENTE
50 PRINT "JEL ORDEN DE LOS DATOS ES: ";
60 INPUT N
70 DELETE S5,R5
80 DIM S5(N),R5(N)
90 FOR I=1 TO N STEP 4
00 PRINT "JDATO(";I;")=";
10 INPUT S5(I)
20 J=I+1
30 PRINT "K          DATO(";J;")=";
40 INPUT S5(J)
50 J=J+1
60 PRINT "K          DATO(";J;")=";
70 INPUT S5(J)
80 J=J+1
90 PRINT "K          DATO(";J;")=";
00 INPUT S5(J)
10 NEXT I
20 PRINT "JEL FACTOR DE CORRELACION R ES : ";
30 INPUT R7
40 PRINT "JY LA RELACION RUIDO SENAL ES : ";
50 INPUT K0
60 PRINT "JDESEA QUE SE PONGA RUIDO A LA SENAL? (SI O NO) ";
70 INPUT Q$
80 IF Q$="S" OR Q$="SI" THEN 1330
90 R5=0
00 PRINT "JLA SENAL SE PROCESA SIN RUIDO"
10 PRINT "J===== "
20 GO TO 1560
30 REM EL RUIDO SE GENERA CON UN NUMERO ALEATORIO QUE GENERA
40 REM          EL COMPUTADOR
50 REM Y SE ALMACENA EN R5 ( VECTOR DE RUIDO )
70 Q1=0
80 FOR I=1 TO N
90 Q1=Q1+S5(I)*S5(I)
00 NEXT I
05 PRINT "LJJSI DESEA RUIDO GAUSSIANO INGRESE LA CANTIDAD DE "
06 PRINT "JJNUMEROS ALEATORIOS UNIFORMEMENTE DISTRIBUIDOS"
07 PRINT "JJA SUMARSE; EN EL CASO DE DESIAR UN RUIDO"
08 PRINT "JJUNIFORMEMENTE DISTRIBUIDO, INGRESE EL (%)"
09 INPUT Q
10 Q2=0
15 PRINT "LJJJEL VECTOR RUIDO GENERADO EN EL COMPUTADOR ES: "
20 FOR I=1 TO N
30 R5(I)=RND(1)-0.5
40 Q2=Q2+R5(I)*R5(I)

```

```
450 NEXT I
452 IF Q>0 THEN 1570
460 O1=O1/N
470 O2=O2/N
480 P0=SQR(O1/O2)
490 FOR I=1 TO N
500 R5(I)=K0*P0*R5(I)
510 NEXT I
520 FOR I=1 TO N STEP 2
530 PRINT @32: USING 1550:I,R5(I),I+1,R5(I+1)
540 NEXT I
550 IMAGE/"  RUIDO (" ,2D," ) =" ,6D.3D,"  RUIDO (" ,2D," ) =" ,6D.3D
560 END
570 PRINT "Ud oindra` en un momento el ruido Gaussiano"
580 O3=0
585 O2=0
590 FOR J=1 TO N
595 R5(J)=0
600 FOR K=1 TO Q
610 R5(J)=R5(J)+RND(1)
620 NEXT K
630 O2=O2+R5(J)
640 NEXT J
650 O2=O2/N
660 FOR I=1 TO N
670 R5(I)=R5(I)-O2
680 O3=O3+R5(I)*R5(I)
690 NEXT I
700 O2=O3
710 GO TO 1460
```

```

00 REM          "FILTRO/COSENO"
10 O2=3
20 GO TO 800
30 REM
40 REM          PROGRAMA PARA HACER FILTRADO POR
50 REM          LA TRANSFORMADA RAPIDA COSENO
60 REM          E. P. N.
70 REM          JAIME ARTURO ERAZO
80 REM EL VECTOR SENAL ES S5
90 REM EL VECTOR RUIDO ES R5
00 REM EL ORDEN DE LOS VECTORES ES N
10 A$="FILCOS"
20 X$=A$
30 OPEN X$;1,"R",0$
40 IF TYP(1)=1 THEN 2990
50 READ #1:D1,D7
60 DELETE G
70 DIM G(D1)
80 READ #1:G
90 IF D7=R7 AND D1=N THEN 1210
00 GO TO 1140
10 REM REN SE LOCALIZA LA MATRIZ FILTRO EN EL ARCHIVO
20 M=N
30 P1=51
40 PRINT @P1:"LJJJJ          FILTRADO DE WIENER POR LA TRANSFORMADA"
50 PRINT @P1:"J          C O S E N O"
60 PRINT @P1:"=====
70 PRINT @P1:"JJEL NUMERO DE DATOS ES : ";M
80 PRINT @P1:"JJLA SENAL ES :JJ",S5
90 PRINT @P1:"JJY EL RUIDO ;JJ",R5
00 PRINT @P1:"LA RELACION RUIDO/SENAL ES ;";K0
10 PRINT @P1:"EL FACTOR DE CORRELACION ES ;";R7
20 REM AQUI: TRANSFORMADA RAPIDA COSENO          INICIO
30 N=M/2
40 DELETE F,R,S,T,W,U,C,V,CB
50 DIM F(M),R(N),S(N),T(2),U(2),W(2),C(M),V(M),CB(M)
60 FOR I=1 TO M
70 F(I)=S5(I)+R5(I)
80 NEXT I
90 REM CALCULO DE LA TRANSFORMADA COSENO DIRECTA
00 FOR I=1 TO N
10 J=I*2-1
20 V(I)=F(J)
30 NEXT I
40 K=N+1
50 FOR I=K TO M
60 K=I-1
70 L=2*(M-K)
80 V(I)=F(L)
90 NEXT I

```

```

00 FOR I=1 TO N
10 J=2*I
20 K=J-1
30 R(I)=V(K)
40 S(I)=V(J)
50 NEXT I
60 REM TRANSFORMADA COSENO UTILIZANDO FOURIER      : : : : * : : : :
70 P=PI
80 SET RADIANS
90 N2=N/2
00 N1=N-1
10 J=1
20 FOR I=1 TO N1
30 IF I=>J THEN 1700
40 T(1)=R(J)
50 T(2)=S(J)
60 R(J)=R(I)
70 S(J)=S(I)
80 R(I)=T(1)
90 S(I)=T(2)
00 K=N2
10 IF K=>J THEN 1750
20 J=J-1
30 K=K/2
40 GO TO 1710
50 J=J+K
60 NEXT I
70 Q=1
80 L1=0
90 Q=Q*2
00 L1=L1+1
10 IF Q<N THEN 1790
20 FOR L=1 TO L1
30 L2=2^L
40 L3=L2/2
50 U(1)=1
60 U(2)=0
70 W(1)=COS(P/L3)
80 W(2)=SIN(P/L3)
90 FOR J=1 TO L3
00 FOR I=J TO N STEP L2
10 D=I+L3
20 T(1)=R(D)*U(1)-S(D)*U(2)
30 T(2)=S(D)*U(1)+R(D)*U(2)
40 R(D)=R(I)-T(1)
50 S(D)=S(I)-T(2)
60 R(I)=R(I)+T(1)
70 S(I)=S(I)+T(2)
80 NEXT J
90 A1=U(1)*W(1)-U(2)*W(2)

```

```

000 A2=U(1)*W(2)+U(2)*W(1)
010 U(1)=A1
020 U(2)=A2
030 NEXT J
040 NEXT L
050 REM FIN DEL PROCESO FOURIER EN COSENO DIRECTA
060 REM
070 FOR K=1 TO N
080 L=K-1
090 CO=2*COS(P*L/M)
100 SO=2*SIN(P*L/M)
110 C(K)=R(K)*CO+SO*S(K)
120 J=M-L
130 C(J)=S(K)*CO-R(K)*SO
140 NEXT K
150 REM FIN DE TRANSFORMADA RAPIDA COSENO
160 REM
170 REM TRANSFORMADA COSENO INVERSA
180 REM PRIMERO HACEMOS INGRESAR LOS DATOS A LA COSENO INVERSA
190 FOR I=1 TO M
200 CB(I)=0
210 CB(I)=CB(I)+C(I)*G(I)/(G(I)+K0)
220 NEXT I
230 C=CB
240 FOR K=1 TO N
250 CO=-COS(P*(K-1)/M)
260 SO=SIN(P*(K-1)/M)
270 S(K)=C(K)*SO-C(M-K+1)*CO
280 R(K)=C(K)*CO+C(M-K+1)*SO
290 NEXT K
300 REM TRANSFORMADA COSENOINVERSA UTILIZANDO FOURIER INVERSO
310 J=1
320 FOR I=1 TO N1
330 IF I=>J THEN 2400
340 T(1)=R(J)
350 T(2)=S(J)
360 R(J)=R(I)
370 S(J)=S(I)
380 R(I)=T(1)
390 S(I)=T(2)
400 K=N2
410 IF K=>J THEN 2450
420 J=J-1
430 K=K/2
440 GO TO 2410
450 J=J+K
460 NEXT I
470 Q=1
480 L1=0
490 Q=Q*2

```

```

00 L1=L1+1
10 IF Q<N THEN 2490
20 FOR L=1 TO L1
30 L2=2^L
40 L3=L2/2
50 U(1)=1
60 U(2)=0
70 W(1)=COS(P/L3)
80 W(2)=SIN(P/L3)
90 FOR J=1 TO L3
00 FOR I=J TO N STEP L2
10 D=I+L3
20 T(1)=R(D)*U(1)-S(D)*U(2)
30 T(2)=S(D)*U(1)+R(D)*U(2)
40 R(D)=R(I)-T(1)
50 S(D)=S(I)-T(2)
60 R(I)=R(I)+T(1)
70 S(I)=S(I)+T(2)
80 NEXT I
90 A1=U(1)*W(1)-U(2)*W(2)
00 A2=U(1)*W(2)+U(2)*W(1)
10 U(1)=A1
20 U(2)=A2
30 NEXT J
40 NEXT L
50 REM FIN DEL PROCESO FOURIER EN COSENO INVERSA !!!!!*!!!!:8
60 R=R/M
70 R=-R
80 S=S/M
90 J=0
00 FOR K=1 TO N2
10 L=N-K+1
20 J=J+1
30 F(J)=R(K)
40 J=J+1
50 F(J)=S(L)
60 J=J+1
70 F(J)=S(K)
80 J=J+1
90 F(J)=R(L)
00 NEXT K
10 N=M
20 PRINT @P1:"JJLA ESTIMACION DE LA SENAL ES : "
30 FOR K=1 TO M STEP 2
40 PRINT @P1: USING 2950:K,F(K),K+1,F(K+1)
50 IMAGE /,5X,"*- S (",3D," ) =",5D.2D,20X,"*- S (",3D," ) =",5D.2D
60 NEXT K
70 N$=" C O S E N O "
80 GO TO 4
90 PRINT "LJJJJJJJJJJJJJJJJJJG"

```

```

) REM          "FILTRADO/FOURIER"
) D2=4
) GO TO 800
) REM
) REM          PROGRAMA PARA HACER FILTRADO POR
) REM          LA TRANSFORMADA RAPIDA FOURIER
) REM          E. P. N.
) REM          JAIME ARTURO ERAZO
) REM
) REM EL VECTOR SENAL ES S5
) REM EL VECTOR RUIDO ES R5
) REM EL ORDEN DE LOS VECTORES ES N
) REM
) A$="FILFOU"
) X$=A$
) OPEN X$;1,"R",0$
) IF TYP(1)=1 THEN 2560
) READ #1:D1,D7
) DELETE G
) DIM G(D1)
) READ #1:G
) IF D7=R7 AND D1=N THEN 1250
) GO TO 1150
) REM REN SE LOCALIZA LA MATRIZ FILTRO EN EL ARCHIVO
) REM
) PRINT @P1:"LJJJJ          FILTRADO DE WIENER POR LA TRANSFORMADA DE"
) PRINT @P1:"J                      F O U R I E R"
) PRINT @P1:"===== "
) PRINT @P1:"JJEL NUMERO DE DATOS ES : ";N
) PRINT @P1:"JJLA SENAL ES :JJ",S5
) PRINT @P1:"JJY EL RUIDO :JJ",R5
) PRINT @P1:"LA RELACION RUIDO/SENAL ES :";K0
) PRINT @P1:"EL FACTOR DE CORRELACION ";R7
) REM AQUI: TRANSFORMADA RAPIDA COSENO          INICIO
) DELETE Y,R,S,T,W,U,C8,F
) DIM Y(N),R(N),S(N),T(2),U(2),W(2),C8(N),F(N)
) R=S5+R5
) S=0
) REM CALCULO DE LA TRANSFORMADA DE FOURIER DIRECTA
) P=PI
) SET RADIANS
) N2=N/2
) N1=N-1
) J=1
) FOR I=1 TO N1
) IF I=>J THEN 1510
) T(1)=R(J)
) T(2)=S(J)
) R(J)=R(I)
) S(J)=S(I)
) R(I)=T(1)

```

```

10 S(I)=T(2)
20 K=N2
30 IF K=>J THEN 1560
40 J=J-1
50 K=K/2
60 GO TO 1520
70 J=J+K
80 NEXT I
90 Q=1
100 L1=0
110 Q=Q*2
120 L1=L1+1
130 IF Q<N THEN 1600
140 FOR L=1 TO L1
150 L2=2^L
160 L3=L2/2
170 U(1)=1
180 U(2)=0
190 W(1)=COS(P/L3)
200 W(2)=SIN(P/L3)
210 FOR J=1 TO L3
220 FOR I=J TO N STEP L2
230 D=I+L3
240 T(1)=R(D)*U(1)-S(D)*U(2)
250 T(2)=S(D)*U(1)+R(D)*U(2)
260 R(D)=R(I)-T(1)
270 S(D)=S(I)-T(2)
280 R(I)=R(I)+T(1)
290 S(I)=S(I)+T(2)
300 NEXT I
310 A1=U(1)*W(1)-U(2)*W(2)
320 A2=U(1)*W(2)+U(2)*W(1)
330 U(1)=A1
340 U(2)=A2
350 NEXT J
360 NEXT L
370 REM FIN DEL PROCESO FOURIER DIRECTO *****!!!!
380 FOR I=1 TO N
390 C8(I)=0
400 C8(I)=C8(I)+R(I)*G(I)/(G(I)+K0)
410 NEXT I
420 R=C8
430 FOR I=1 TO N
440 C8(I)=0
450 C8(I)=C8(I)+S(I)*G(I)/(G(I)+K0)
460 NEXT I
470 S=-C8
480 REM TRANSFORMADA DISCRETA DE FOURIER INVERSA
490 J=1
500 FOR I=1 TO N1

```

```

00 IF I=>J THEN 2070
10 T(1)=R(J)
20 T(2)=S(J)
30 R(J)=R(I)
40 S(J)=S(I)
50 R(I)=T(1)
60 S(I)=T(2)
70 K=N2
80 IF K=>J THEN 2120
90 J=J-1
00 K=K/2
10 GO TO 2080
20 J=J+K
30 NEXT I
40 Q=1
50 L1=0
60 Q=Q*2
70 L1=L1+1
80 IF Q<N THEN 2160
90 FOR L=1 TO L1
00 L2=2^L
10 L3=L2/2
20 U(1)=1
30 U(2)=0
40 W(1)=COS(P/L3)
50 W(2)=SIN(P/L3)
60 FOR J=1 TO L3
70 FOR I=J TO N STEP L2
80 D=I+L3
90 T(1)=R(D)*U(1)-S(D)*U(2)
00 T(2)=S(D)*U(1)+R(D)*U(2)
10 R(D)=R(I)-T(1)
20 S(D)=S(I)-T(2)
30 R(I)=R(I)+T(1)
40 S(I)=S(I)+T(2)
50 NEXT I
60 A1=U(1)*W(1)-U(2)*W(2)
70 A2=U(1)*W(2)+U(2)*W(1)
80 U(1)=A1
90 U(2)=A2
00 NEXT J
10 NEXT L
20 REM FIN DEL PROCESO DE LA T, DE FOURIER INVERSA
30 R=R/N
40 S=S/N
50 S=-S
60 PRINT @P1:"JULA ESTIMACION DE LA SENAL ES ; "
70 FOR K=1 TO N STEP 2
80 PRINT @P1: USING 2490:K,R(K),S(K),K+1,R(K+1),S(K+1)
90 IMAGE/,5X,"*-S(",3D,")=",5D,2D,5D,2D,20X,"*-S(",3D,")=",5D,2D,5D,2D

```

```
10 NEXT K
0 N$="          D E    F O U R I E R"
10 FOR I=1 TO N
10 F(I)=SQR(S(I)*S(I)+R(I)*R(I))
10 NEXT I
10 GO TO 4
10 PRINT "LJJJJJJJJJJJJJJJJG"
10 PRINT "G          ENCONTRO FIN DE ARCHIVO,PROCEDE ACTUALIZARLO"
30 CLOSE
10 GO TO 36
```

```

0 REM          "FILTRO/KARLOEVE"
0 D2=5
0 GO TO 800
0 REM
0 REM          PROGRAMA PARA HACER FILTRADO FOR
0 REM          LA TRANSFORMADA KARHUNEN LOEVE
0 REM          E. P. N.
0 REM          JAIME ARTURO ERAZO
0 REM
0 REM EL VECTOR SENAL ES S5
0 REM EL VECTOR RUIDO ES R5
0 REM EL ORDEN DE LOS VECTORES ES N
0 A$="DATOG"
0 X#=A$
0 OPEN X#;1,"R",0$
0 IF TYF(1)=1 THEN 1710
0 READ #1:D1,D7
0 DELETE G
0 DIM G(D1)
0 READ #1:G
0 IF D7=R7 AND N=D1 THEN 1210
0 GO TO 1140
0 REM REN SE LOCALIZA LA MATRIZ FILTRO EN EL ARCHIVO
20 P1=51
30 PRINT @P1:"LJJJJ          FILTRADO DE WIENER POR LA TRANSFORMADA"
40 PRINT @P1:"J          K A R H U N E N   L O E V E"
50 PRINT @P1:"=====*"
60 PRINT @P1:"JJEL NUMERO DE DATOS ES : ";N
70 PRINT @P1:"JJLA SENAL ES :JJ",S5
80 PRINT @P1:"JJY EL RUIDO :JJ",R5
90 PRINT @P1:"LA RELACION RUIDO/SENAL ES :";K0
00 PRINT @P1:"LA COVARIANCIA ES :";R7
0 REM AQUI: TRANSFORMADA KARHUNEN LOEVE          INICIO
20 DELETE A,F1,F,C
30 DIM A(N,N),F1(N,N),F(N),C(N)
40 F=S5+R5
50 F=F1
60 SET RADIANS
70 N1=N+1
80 FOR I=1 TO N
90 K=I+1
00 F1(I,I)=SQR(2/N1)*SIN(I*I*P/N1)
10 IF K>N THEN 1470
20 FOR J=K TO N
30 F1(I,J)=SQR(2/N1)*SIN(I*J*P/N1)
40 F1(J,I)=F1(I,J)
50 NEXT J
60 NEXT I
70 REM FIN DE LOS CALCULOS DE LA MATRIZ K,L.
80 FOR J=1 TO N
90 C(J)=0

```

```

10 FOR K=1 TO N
20 C(J)=C(J)+F(K)*F1(J,K)
30 NEXT K
40 NEXT J
50 REM SE TRANSFORMO EL VECTOR DE ENTRADA POR KARHUNEN LOEVE (FIN)
60 FOR I=1 TO N
70 C(I)=C(I)*G(I)/(G(I)+K0)
80 NEXT I
90 FOR J=1 TO N
10 F(J)=0
110 FOR K=1 TO N
120 F(J)=F(J)+C(K)*F1(J,K)
130 NEXT K
140 NEXT J
150 PRINT @P1:"JULA ESTIMACION DE LA SENAL ES : "
160 FOR K=1 TO N STEP 2
170 PRINT @P1: USING 1670:K,F(K),K+1,F(K+1)
180 IMAGE /,5X,"*- S (",3D," ) =",5D,2D,20X,"*- S (",3D," ) =",5D,2D
190 NEXT K
200 N$="      K A R H U N E N L O E V E"
210 GO TO 4
220 PRINT "LJJJJJJJJJJJJJJJJJJJJJJJJJJJJG"
230 PRINT "LLEGO AL FIN DEL ARCHIVO, SIGUE EL PROCESO DE ACTUALIZACION"
240 CLOSE
250 GO TO 28

```

```

00 REM          *GRAFICO*
10 O2=6
20 GO TO 800
30 REM          PROGRAMA PARA HACER GREFICOS DE
40 REM          DATOS Y RESULTADOS
50 REM          E. P. N.
60 REM          JAIME ARTURO ERAZO
70 REM
00 E$="FILTRADO POR LA TRANSFORMADA"
50 I$="FACTOR DE CORRELACION = "
60 J$="RELACION RUIDO/SENAL = "
20 GOSUB 6000
30 AXIS @1:1,1,0,0
35 MOVE @1:1,S5(1)
40 FOR I=1 TO N
50 DRAW @1:I,S5(I)
60 NEXT I
70 GOSUB 6000
10 PRINT "PONGA EL COLOR ROJO Y APLASTE RETURN PARA CONTINUAR";
20 INPUT R$
30 MOVE @1:1,S5(1)+R5(1)
40 FOR I=1 TO N
50 DRAW @1:I,S5(I)+R5(I)
60 NEXT I
80 PRINT "PONGA EL COLOR VERDE Y APLASTE RETURN PARA CONTINUAR";
90 INPUT R$
00 MOVE @1:1,F(1)
10 FOR I=1 TO N
20 DRAW @1:I,F(I)
30 NEXT I
40 PRINT "PONGA EL COLOR NEGRO Y RETURN PARA CONTINUAR"
50 INPUT R$
60 MOVE @1:1,25
65 PRINT @1,7:
70 PRINT @1:"F I L T R A D O   P O R   T R A N S F O R M A D A"
75 MOVE @1:1,20
77 PRINT @1:"          ";N$
80 PRINT @1,17:0.8*1.792,0.5*2.816
81 MOVE @1:0,3
82 PRINT @1:
83 PRINT @1:"A"
84 PRINT @1:"JM"
85 PRINT @1:"JF"
86 PRINT @1:"JL"
87 PRINT @1:"JI"
88 PRINT @1:"JT"
89 PRINT @1:"JU"
91 PRINT @1:"JD"
92 MOVE @1:1,-17
00 PRINT @1:I$,R7

```

```
5 MOVE @1:1,-18
0 PRINT @1:J$,K0
0 F6=N/100
0 FOR I=-15 TO 15 STEP 2
0 MOVE @1:F6,I
0 PRINT @1:I
0 NEXT I
0 FOR I=1 TO N
0 MOVE @1:I,-1
0 PRINT @1:I
0 NEXT I
0 MOVE @1:16+N/50,0.5
0 PRINT @1:'[n]'
0 GO TO 4
0 WINDOW 0,N,-15,15
0 VIEWPORT 6*1.792,150-8*1.792,4*2.816,100-10*2.816
0 RETURN
```

```

50 REM PROGRAMA PARA ACTUALIZAR EL ARCHIVO DE FILTRO KARHUNEN LOEVE
10 O2=7
20 GO TO 800
30 REM PROGRAMA PARA HACER LAS MATRICES DEL FILTRO
70 R=R7
00 DELETE A,F1,F,C,G
10 DIM A(N,N),F1(N,N),F(N,N),C(N,N),G(N)
20 F=PI
30 SET RADIANS
40 N1=N+1
50 FOR I=1 TO N
60 K=I+1
70 F1(I,I)=SQR(2/N1)*SIN(I*I*F/N1)
80 IF K>N THEN 1240
90 FOR J=K TO N
00 F1(I,J)=SQR(2/N1)*SIN(I*J*F/N1)
10 F1(J,I)=F1(I,J)
20 NEXT J
30 NEXT I
40 REM FIN DE LOS CALCULOS DE LA MATRIZ K.L.
50 REM VER F1
60 FOR I=1 TO N
70 FOR J=I TO N
80 A(I,J)=R^(J-I)
90 A(J,I)=A(I,J)
00 NEXT J
10 NEXT I
20 REM VER A MATRIZ COVARIANCIA
30 FOR I=1 TO N
40 FOR J=1 TO N
50 F(I,J)=0
60 FOR K=1 TO N
70 F(I,J)=F(I,J)+F1(I,K)*A(K,J)
80 NEXT K
90 NEXT J
00 NEXT I
10 FOR I=1 TO N
20 FOR J=1 TO N
30 C(I,J)=0
40 FOR K=1 TO N
50 C(I,J)=C(I,J)+F(I,K)*F1(K,J)
60 NEXT K
70 NEXT J
80 NEXT I
90 FOR I=1 TO N
00 G(I)=C(I,I)
10 NEXT I
20 OPEN "DATOG";1,"U",0$
30 WRITE #1:N,R,G
40 CLOSE

```

```

0 REM PROGRAMA PARA ACTUALIZAR EL ARCHIVO DE FILTRO FOURIER
0 D2=9
0 GO TO 800
0 REM PROGRAMA PARA HACER LAS MATRICES DEL FILTRO
0 DELETE A,F1,F,C,G
0 DIM A(N,N),F1(N,N),F(N,N),C(N,N),G(N)
0 P=PI
0 SET RADIANS
0 REM SE GENERO LA MATRIZ DE COVARIANCIA
0 FOR I=1 TO N
0 FOR J=I TO N
0 A(I,J)=R7^(J-I)
0 A(J,I)=A(I,J)
0 NEXT J
0 NEXT I
0 REM VER A MATRIZ COVARIANCIA
0 DIM Y(N),R(N),S(N),T(2),U(2),W(2)
05 FOR T5=1 TO N
06 FOR T6=1 TO N
08 R(T6)=A(T5,T6)
09 NEXT T6
10 S=0
10 REM CALCULO DE LA TRANSFORMADA DE FOURIER DIRECTA
10 P=PI
10 SET RADIANS
10 N2=N/2
10 N1=N-1
10 J=1
10 FOR I=1 TO N1
10 IF I=>J THEN 1790
10 T(1)=R(J)
10 T(2)=S(J)
10 R(J)=R(I)
10 S(J)=S(I)
10 R(I)=T(1)
10 S(I)=T(2)
10 K=N2
10 IF K=>J THEN 1840
10 J=J-1
10 K=K/2
10 GO TO 1800
10 J=J+K
10 NEXT I
10 Q=1
10 L1=0
10 Q=Q*2
10 L1=L1+1
10 IF Q<N THEN 1880
10 FOR L=1 TO L1
10 L2=2^L

```

```

) L3=L2/2
) U(1)=1
) U(2)=0
) W(1)=COS(F/L3)
) W(2)=SIN(F/L3)
) FOR J=1 TO L3
) FOR I=J TO N STEP L2
) D=I+L3
) T(1)=R(D)*U(1)-S(D)*U(2)
) T(2)=S(D)*U(1)+R(D)*U(2)
) R(D)=R(I)-T(1)
) S(D)=S(I)-T(2)
) R(I)=R(I)+T(1)
) S(I)=S(I)+T(2)
) NEXT I
) A1=U(1)*W(1)-U(2)*W(2)
) A2=U(1)*W(2)+U(2)*W(1)
) U(1)=A1
) U(2)=A2
) NEXT J
) NEXT L
) REM FIN DEL PROCESO FOURIER DIRECTO *****:!!!
) FOR T6=1 TO N
) F(T5,T6)=R(T6)
) F1(T5,T6)=S(T6)
) NEXT T6
) NEXT T5
) C=F MPY F
) A=F1 MPY F1
) C=C+A
) R2=R7*R7
) G(1)=C(1,1)/((1-R2)*N)
) FOR I=2 TO N-1
) G(I)=C(I,I)*(1+R2)/((1-R2)*N)
) NEXT I
) G(N)=C(N,N)/((1-R2)*N)
) OPEN "FILFOU";1,"U",0#
) WRITE #1:N,R7,G
) CLOSE
) GO TO 16

```

```

) REM PROGRAMA PARA ACTUALIZAR EL ARCHIVO DE FILTRO COSENO
) O2=8
) GO TO 800
) REM PROGRAMA PARA HACER LAS MATRICES DEL FILTRO
) DELETE A,F1,F,C,G
) DIM A(N,N),F1(N,N),F(N,N),C(N,N),G(N)
) P=PI
) SET RADIANS
) FOR I=1 TO N
) FOR J=1 TO N
) F1(I,J)=COS(I*PI*(J-1)/N)
) NEXT J
) NEXT I
) REM FIN DE LOS CALCULOS DE LA MATRIZ COSENO
) REM VER F1
) FOR I=1 TO N
) FOR J=I TO N
) A(I,J)=R7^(J-I)
) A(J,I)=A(I,J)
) NEXT J
) NEXT I
) REM VER A MATRIZ COVARIANCIÁ
) FOR I=1 TO N
) FOR J=1 TO N
) F(I,J)=0
) FOR K=1 TO N
) F(I,J)=F(I,J)+F1(I,K)*A(J,K)
) NEXT K
) NEXT J
) NEXT I
) FOR I=1 TO N
) FOR J=1 TO N
) C(I,J)=0
) FOR K=1 TO N
) C(I,J)=C(I,J)+F(I,K)*F1(K,J)
) NEXT K
) NEXT J
) NEXT I
) FOR I=1 TO N
) G(I)=C(I,I)
) NEXT I
) OPEN "FILCOS";1,"U",0$
) WRITE #1:N,R7,G
) CLOSE
) GO TO 12

```

B I B L I O G R A F I A

- [1] Marcelo Rodas Pozo, Transformada Rápida de Fourier, Tesis de Grado, EPN. 1977.
- [2] A.V. Oppenheim and R. W. Schaffer, Digital Signal Processing Englewood Cliffs, NJ: Prentice-Hall, 1975.
- [3] J.M. Tribolet and R.E. Crochiere, "A Vocoder driven adaptation strategy for low-bit rate adaptive transform coding of speech", in Proc. 1978 Int. Conf. Digital Signal Processing, Florence, Italy, Aug. 30-Sep. 2, 1978.
- [4] I. Markhouf, "A Fast Cosine Transform in One and Two Dimensions", IEEE. Trans. Vol. ASSP-28 Nº 1 pp. 27-34. Feb. 1980.
- [5] J. Makhoul. "Speaker - Machine interaction in automatic speech recognition", Res. Lab. Electron, Massachusetts. Inst. Tech., Cambridge, M.A. Tech. Rep. 480, Dec. 15, 1979; También, Ph.D. dissertation, Dept. Elec. Eng., Massachusetts, Inst. Tech., Cambridge., M.A. 1970.
- [6] W.B. Davenport, Jr., and W.L. Root, Random Signals and Noise. New York: McGraw-Hill 1968.
- [7] J. Pearl, "Basis-restricted transformations and performance measures for spectral representation" IEEE Trans. Infor. Theory (Corresp.), Vol. IT-17, pp. 751-752, Nov. 1971.
- [8] W.K. Pratt "Linear and nonlinear filtering in the Walsh domain" presented at the simp. Appl, Walsh Functions, Washington, D.C. Apr. 1971.
- [9] W.K. Pratt "Generalized Wiener Filtering Computation techniques" IEEE Trans. COMPUTERS, Vol C-21, pp. 636-641, July 1972.

- [10] "Walsh processing and random signals", Presented at the Symp. Appl. Walsh Functions, Washington D.C., Apr. 1971.
- [11] M.J. Narasimha y A.M. Peterson, "On the computation of the discrete coseno transform", IEEE Trans. Commun., Vol COM 26, pp. 934-936, Junio 1978.
- [12] E.O. Brigham, The Fast Fourier Transform. Englewood Cliffs, N.J.: Prentice Hall, 1974, pp. 166-169.
- [13] The Karhunen-Loève Transform and Block Quantization, Proceedings of the IEEE, Vol. 69, Nº 3. pp. 364. Marzo 1981.
- [14] U. Grenander and G. Szego, Teopltitz Forms and Their Applications. Berkeley: Univ. California Press, 1958.
- [15] O. Teopltitz, "Zur Theorie der Quadratischen und Bilinearen Formen von Unendlichvielen Veranderlichen, I. Teil: Theorie del L. Formen", Math. Ann., Vol. 70, 1911.
- [16] A.K. Jain, "A fast Karhunen-Loeve transform for finite discrete images", in Proc. Nat. Electron cont., Chicago, II, Oct. 1974.
- [17] A.K. Jain, "A Fast Karhunen-Loeve transform for a class of random process", IEEE Trans. Commun. Sep. 1974.
- [18] A.K. Jain, "Image coding via a nearest neighbors image model", IEEE Trans. Commun. Vol. COM-23, pp. 318-331. Mar. 1975.