



## PROCESAMIENTO DIGITAL DE SEÑALES SÍSMICAS CON MATLAB

César Omar Jiménez Tintaya.<sup>a\*</sup>

<sup>a</sup>CNDG – Sismología, Instituto Geofísico del Perú.

---

### Abstract

The purpose of this work is to implement an algorithm to identify the seismic waves parameters: P phase, S phase, amplitude, period and duration. For this, digital signal processing and the programming language Matlab will be applied to perform the automatic reading of seismic signal parameters. As mathematical tools, the discrete Fourier transform (fft) to study the frequency components of the signal and the Hilbert transform to obtain the envelope signal, will be used. A graphical user interface in Matlab was designed and implemented for reading (with functions of automatic reading or manual reading) seismic signals with three components. In this way, it makes easy the routine of reading the seismic signals within a great set. The results of the automatic algorithm agree with that of a manual operator for a great Signal/Noise ratio.

PACS: 91.30.-f; 91.30.Ab; 07.50.Qx

**Keywords:** seismology, signal processing, Matlab.

---

### Resumen

El objetivo del presente trabajo es implementar un algoritmo para la identificación de los parámetros de la onda sísmica como son: las fases P y S, amplitud, periodo y duración. Para esto se aplicará el procesamiento digital de señales y el lenguaje de programación Matlab para realizar la lectura automática de dichos parámetros. Como herramientas matemáticas se utilizará la transformada discreta de Fourier para estudiar las componentes frecuenciales de la señal y la transformada de Hilbert para obtener la envolvente de la señal. Se diseñó e implementó una interfaz gráfica de usuario en Matlab para la lectura (con funciones de lectura automática o lectura manual) de señales sísmicas con tres componentes. De esta manera, se logra agilizar la rutina al realizar la lectura de las señales sísmicas dentro de un gran conjunto. Los resultados del algoritmo automático concuerdan bien con los de un operador manual para una relación Señal/Ruido grande.

**Palabras claves:** sismología, procesamiento de señales, Matlab.

---

### 1. Introducción

La Tierra no es un cuerpo estático sino que ha estado y está sujeta a una continua actividad, es decir, la Tierra es un sistema dinámico. Esto se debe a la interacción de las fuerzas geológicas del interior de la Tierra. La corteza terrestre está constituida por las llamadas placas tectónicas, las cuales, en su interacción, producen los terremotos y volcanes.

El Cinturón Sísmico del Pacífico es la principal zona de peligro sísmico mundial, ya que casi todas las zonas de subducción presentes en la Tierra se ubican alrededor del océano Pacífico, al igual que más de las tres cuartas partes de todos los volcanes activos y extinguidos. El Cinturón Sísmico del Pacífico se extiende a lo largo de la costa occidental de América, las islas Aleutianas, la costa oriental de Asia y las islas del Japón.

Los márgenes de subducción marcan aquellas

zonas en las que las placas convergen unas con otras. Este movimiento obliga a una de ellas a introducirse por debajo de la otra, como resultado de esto la litósfera se consume o se destruye [1].

Como producto de esta interacción entre placas, se generan los terremotos, que consiste en el movimiento de la superficie terrestre debido a la propagación de las ondas sísmicas. Estas ondas sísmicas pueden ser de varios tipos, siendo las principales las ondas de volumen: P y S (Figura 1). Las ondas P o primarias son ondas longitudinales de compresión. Las ondas S o secundarias son ondas transversales de cizalla.

La presencia de estas ondas son detectadas por los instrumentos llamados sismógrafos, que básicamente consisten en un sistema mecánico: masa, resorte y amortiguador (un sistema de segundo orden) y un sistema de grabación de datos, ya sea en un medio analógico (aguja sobre tambor con papel ahumado) o

---

\* E-mail: [cjimenezt@unmsm.edu.pe](mailto:cjimenezt@unmsm.edu.pe)

mediante un medio digital (PC con conversor analógico digital).

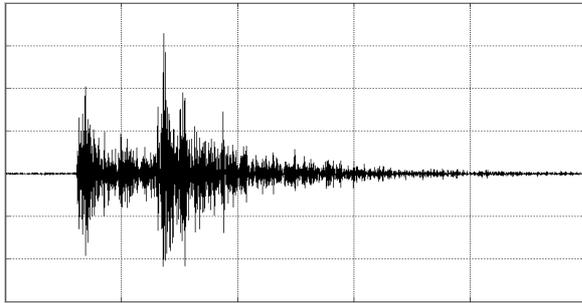


Fig. 1. Señal sísmica: el primer tren corresponde a las ondas P y el segundo, de mayor amplitud, a las ondas S.

Con los tiempos de arribo o llegada de las ondas P y S, y la duración del sismo para 3 estaciones sísmicas como mínimo, es posible calcular la localización del hipocentro, la magnitud y el tiempo origen del terremoto. Como una estimación, también puede utilizarse una estación sísmica con 3 componentes, siempre y cuando las fases P sean impulsivas para un sismo local.

**2. Señales y Sistemas**

Una señal es una variable física que contiene o transporta información. Algunos tipos de señales son: tensión eléctrica, sonido, imagen, temperatura captada por un sensor, movimiento, etc. Nos interesan especialmente las señales que varían en el tiempo. No todas las señales varían en el tiempo, por ejemplo: una imagen fotográfica. Las señales variables en el tiempo pueden representarse mediante una función del tiempo  $y(t)$ . Estas señales pueden ser de dos tipos: continuas o discretas [2].

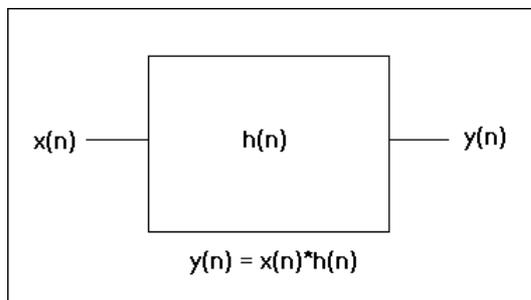


Fig. 2. Modelo de un sistema:  $x(n)$  es la entrada,  $y(n)$  es la salida y  $h(n)$  representa la función de transferencia.

Un sistema es un conjunto interconectado de elementos que procesan una señal. Se caracteriza por tener una o más entradas y una o más salidas (Figura

2). En un sistema lineal se cumple el principio de superposición. Muchos sistemas lineales pueden ser descritos por ecuaciones diferenciales. Todo sistema está caracterizado por su “función de transferencia” (o función de Green), que viene a ser la respuesta del sistema cuando la entrada es un impulso unitario. En forma matemática, la salida del sistema será igual a la convolución de la señal de entrada por la función de transferencia del sistema [3]:

$$y(n) = x(n) * h(n) \tag{1}$$

Si  $x(n) = \delta(n)$  entonces:

$$y(n) = \delta(n) * h(n) = h(n)$$

$h(n)$ : respuesta impulsional

El procesamiento digital de señales implica haber expresado previamente la magnitud de la señal en forma numérica. Cada número, así obtenido, representa la magnitud de la señal en un instante determinado. Entonces, tendremos una sucesión de muestras de la señal en lugar de la señal completa. Reemplazaremos así la señal continua por la señal discreta, en la cual solo interesan los valores en instantes discretos. En la práctica el muestreo se logra multiplicando la señal continua por un tren de impulsos unitarios, tal como se aprecia en la Figura 3.

El teorema del muestreo establece que si una señal se muestrea de manera que se cumpla la condición de Nyquist ( $\omega_s > 2\omega_m$ ), es decir, que la frecuencia de muestreo tiene que ser mayor que la máxima frecuencia contenida en la señal, entonces es posible recuperar la señal. Generalmente, las señales sísmicas contienen componentes de frecuencias de 1 a 10 Hz, por lo que la frecuencia de muestreo debe ser de 20 Hz como mínimo [4].

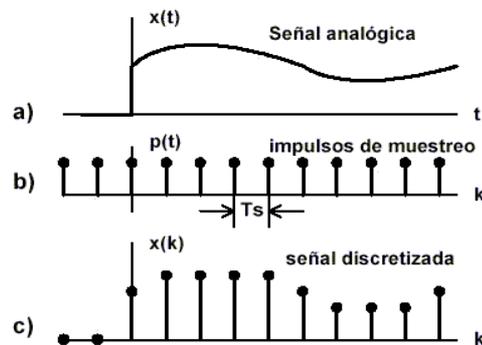


Fig. 3. El proceso del muestreo

**2.1 El Lenguaje Matlab**

Matlab (Matrix Laboratory) es un lenguaje de

programación técnico-científico que básicamente trabaja con variables vectoriales y matriciales. Es fácil de utilizar debido a que contiene varias cajas de herramientas con funciones incorporadas (toolbox de procesamiento de señales, teoría de control, wavelets, matemática simbólica, etc.). La versión que hemos utilizado es la 7.0. Es posible trabajar en programación estructurada o en programación orientada a objetos con una interfaz gráfica de usuario. Es casi un estándar en la programación en ciencias e ingeniería. Es apropiado para el caso de las señales sísmicas donde la frecuencia de muestreo es relativamente baja (de 20 Hz a 100 Hz). Ejemplo de programa en Matlab:

```
load sismo.txt % carga el archivo de datos
y = sismo; % declara una variable
N = length(y); % longitud del vector "y"
Fs = 20; % frecuencia de muestreo
t = (0:N-1)/Fs; % declara la variable tiempo
plot(t,y), grid % gráfica de la señal
```

El resultado de este pequeño segmento de programa está en la Figura 1. Solo se han utilizado 6 líneas. Si esto se realizara en otro lenguaje como Pascal o C++, el programa sería mucho más largo. La potencia de Matlab radica en su simplicidad.

El inconveniente de Matlab es que se torna lento si la longitud del vector (o matriz) es relativamente grande. Se debe evitar el uso del bucle "for k = 1:N" para N muy grande, debido a que demora el cálculo, el bucle "for" es extremadamente ineficiente en Matlab, ya que se trata de un lenguaje interpretado. Sin embargo, se puede vectorizar las operaciones de cálculo, lo cual agiliza el cómputo. El proceso de convertir un bucle "for" en una operación matricial o vectorial se denomina vectorización [5].

## 2.2 Herramientas Matemáticas

La transformada de Fourier es una herramienta matemática muy importante en el procesamiento de señales que sirve para transformar una señal del dominio del tiempo al dominio de la frecuencia. Para el caso continuo tenemos:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad (2)$$

y para el caso discreto:

$$X(n) = \sum_{k=0}^{N-1} x(k)e^{-j2\pi nk/N} \quad (3)$$

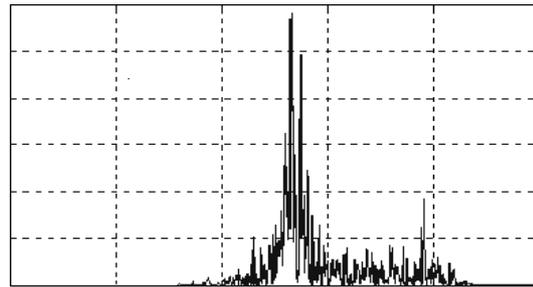


Fig. 4. Señal sísmica en el dominio de la frecuencia o espectro de frecuencias.

El espectro de frecuencias de una señal (Figura 4) nos permitirá saber qué componentes frecuenciales están presentes. Esto servirá para aplicar un filtro adecuado si fuera necesario. El código en Matlab para obtener el espectro de frecuencias de una señal es:

```
Y = fft(y,N); % N: longitud del vector y
Pyy = Y.*conj(Y)/N;
f = (Fs)*(0:N/2-1)/N;
plot(f,Pyy(1:N/2)), grid
```

La transformada de Hilbert es otra herramienta matemática importante que permite obtener la envolvente de una señal. Dicha transformada no cambia la representación de la señal de un dominio a otro, sino que simplemente obtiene de ella otra función [6]:

$$H(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{f(u)}{u-t} du = -\frac{1}{\pi t} * f(t) \quad (4)$$

Si obtenemos el espectro de Fourier de la ecuación (4) como:

$$\mathfrak{F}\left\{-\frac{1}{\pi t}\right\} = i \operatorname{sgn}(s)$$

La transformada de Hilbert equivale a dejar las amplitudes de las componentes espectrales de la señal con la misma amplitud, pero desfasadas por  $\pm \pi/2$ , según el signo de la variable  $s$ .

El proceso del filtrado consiste en eliminar, atenuar o amplificar ciertas componentes frecuenciales de una señal. Por ejemplo, la Figura 5 muestra una señal contaminada con ruido de baja frecuencia, la cual debe ser tratada con un filtro pasa-alta para eliminar dicha componente frecuencial. El resultado del filtrado es la señal de la Figura 6.

Según su aplicación, un filtro puede ser

clasificado como pasa bajo, pasa alto, pasa banda o elimina banda. El filtro más utilizado en sismología es el de Butterworth, debido a su pequeña banda de transición y a la ausencia de ondulaciones o “ripple” en la banda de paso [7]. El rango de frecuencia del ruido sísmico para sismos locales de pequeña y moderada magnitud está comprendido entre 0.1 Hz a 1 Hz. Ejemplo de un filtro “pasa alta” en Matlab:

```
fcorte=1; %elimina debajo de 1Hz
[b, a] = butter(5,fcorte/(Fs/2),'high');
yf = filtfilt(b,a,y);
plot(t,y,'red'), grid on, zoom xon
```

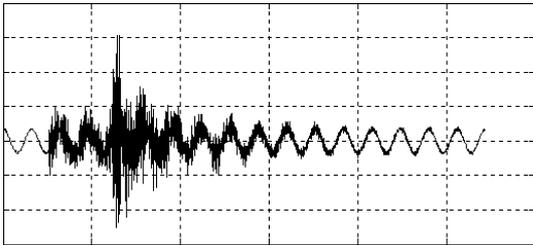


Fig. 5. Señal sísmica contaminada con ruido.

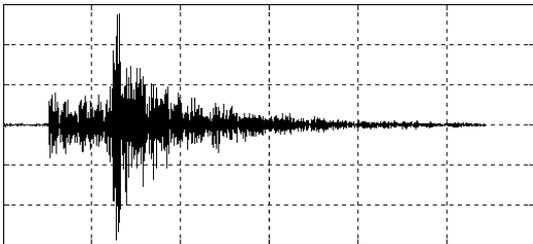


Fig. 6. Señal sísmica anterior después del filtrado.

### 3. Procedimiento

1. Realizar el cambio de formato del archivo de datos (SAC o SUDS) a formato ASCII. Matlab solo lee formato Ascii o su propio formato \*.mat.
2. Utilizar el entorno de programación Matlab. Recordemos que este programa lee las variables como vectores o matrices.
3. Vectorizar la variable tiempo ( $t$ ) y la señal sísmica ( $y$ ).
4. Aplicar una transformada rápida de Fourier a toda la señal y obtener el espectro de frecuencias, observar si hay frecuencias no deseadas (en el rango de 0 a 1 Hz) para luego aplicar un filtro si fuere necesario.
5. La determinación de la fase P se realiza analizando la envolvente de la señal (normalizada a uno, es decir la amplitud máxima de la señal será igual a uno):

$$y_2(t) = \sqrt{y(t)^2 + y_1(t)^2} \quad (5)$$

donde  $y_1(t)$  es la transformada de Hilbert de la señal  $y(t)$ .

Luego, se emplea el siguiente algoritmo de comparación:

- a) Se elige un valor umbral\_1.
- b) Si  $y_2 > \text{umbral}_1$ , se lee el tiempo  $tp$ .
- c) Se busca la posición donde la derivada de  $y_2$  sea negativa y donde  $y_2 < \text{umbral}_2$ .
- d) Se lee el nuevo tiempo  $tp$ .

6. El cálculo de la fase S se realiza empleando el mismo algoritmo anterior pero con diferentes condiciones para umbral\_1 y umbral\_2.

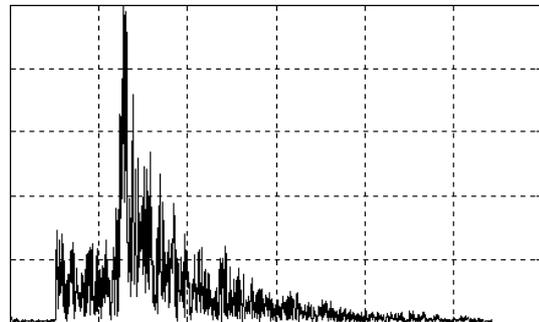


Fig. 7. Envolvente de la señal

7. La amplitud de la onda S estará relacionada con el máximo valor del vector (en Matlab) o señal  $y(t)$ , dentro de los 5 segundos siguientes al tiempo  $ts$ .
8. Se aplica una transformada rápida de Fourier (fft) en torno a la posición del máximo valor de  $y(t)$  para calcular el espectro de frecuencias. El periodo será igual a la inversa de la frecuencia correspondiente al pico máximo.
9. Cálculo de la duración: el cálculo es análogo al de la fase P o S, pero la búsqueda empieza desde el final y hacia atrás. Se lee el tiempo  $td$ . La duración será la diferencia:  $D = td - tp$ .

La interfaz gráfica de usuario (GUI) se realiza con el comando “guide” de Matlab, el cual crea un archivo \*.fig (“fig file”), en el cual pueden insertarse ejes, botones, deslizadores, texto de edición, menús, etc. Cada uno de los cuales tiene ciertas propiedades y realiza una función determinada (“Callback Function”). Por ejemplo: para filtrar la señal, simplemente se hace click en el botón <Filtro>.

#### 4. Resultados

Se realizó el análisis de la señal de la Figura 1 mediante el presente algoritmo y luego se realizó la lectura manual por un operador humano, los resultados son muy cercanos, tal como se indica en la Tabla 1. Todas las unidades están en segundos, excepto para la amplitud, que está en unidades arbitrarias. Este algoritmo ha sido utilizado por el personal de Sismología del Instituto Geofísico del Perú (IGP) para procesar grandes cantidades de señales sísmicas con resultados satisfactorios.

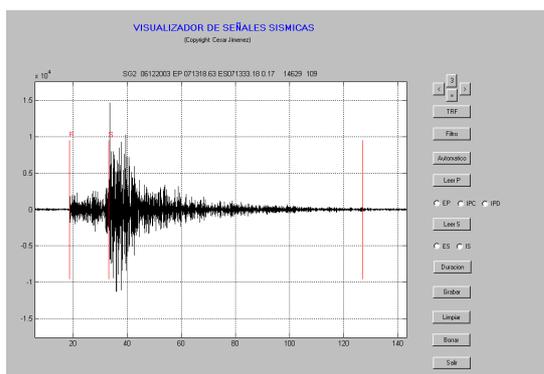


Fig. 8. Interfaz gráfica de usuario del software GRAFSAC.

Aparentemente, existe cierta diferencia en la duración de la señal; sin embargo, lo que realmente interesa para el cálculo de la magnitud del sismo es el logaritmo de la duración, con lo cual se reduciría el error aparente.

Tabla 1. Resultado de la lectura de fases y parámetros.

	Manual	Auto	Delta
$T_p$ (s)	22.76	22.75	0.01
$T_s$ (s)	40.45	40.35	0.10
Dur. (s)	109	118	9
Periodo (s)	0.18	0.18	0
Amp. (ua)	5732	5731	1

#### 5. Observaciones

Si la señal estuviera ruidosa se le debe aplicar un filtro digital para eliminar el ruido, dentro de ciertas condiciones. Esto mejorará la señal, antes de ejecutar el programa.

El presente algoritmo es más eficaz si las fases son relativamente nítidas y si el nivel de ruido es mucho menor que el nivel de la señal sísmica.

Es posible determinar la polaridad de las fases P y S. Posteriormente se incluirá el respectivo algoritmo en el programa.

Para un sismo muy grande donde la señal de la fase S estuviera saturada, el presente algoritmo no sería aplicable. Se tendría que modificar el método de cálculo de la fase S. Por ejemplo, podría utilizarse la transformada wavelet.

#### 6. Conclusiones

Se presenta un nuevo método para la determinación automática del tiempo de arribo de las fases (P y S) y parámetros de una señal sísmica. El algoritmo busca un valor preliminar de  $t_p$  y luego realiza una nueva búsqueda que converge a un valor más exacto de  $t_p$ .

Para un buen cálculo, las fases deben ser relativamente nítidas y el nivel de ruido debe ser menor que el nivel de la señal sísmica.

El algoritmo es rápido y eficaz para señales de sismos locales y regionales.

El programa se utilizó para procesar grandes cantidades de eventos sísmicos con resultados satisfactorios. Lo que antes un operador humano demoraba un mes, ahora es posible hacerlo en un día.

El formato de salida de los datos puede ser cambiado de acuerdo al usuario. El algoritmo es susceptible de ser mejorado.

#### 7. Código Fuente Núcleo

```
% Calculo de las fases, periodo y la % % %
% amplitud de una onda sismica % %
fprintf('%s\n', ' CALCULO DE FASES, %
%AMPLITUD Y PERIODO')
fprintf('\n')
fprintf('%s\n', ' DE ONDAS SISMICAS')
fprintf('\n')
load sismo.txt % Cargar archivo en %
t = sismo(:,1); % Vector tiempo
y = sismo(:,2); % Vector señal sismica
N = length(y); % longitud del vector
Fs = 20;
T = 1/Fs; % Intervalo de muestreo
[A,I]=max(abs(y)); % I = posicion del
%maximo
y1 = hilbert(y);
y2 = sqrt(y.*y+y1.*conj(y1));
A2 = max(y2);
y2 = y2/A2; % envolvente
comp = mean(y2(1:60));
for n = 10:I
    if y2(n) > 5.9*comp
        n;
        break
    end
end
for k = 1:n
    if y2(m-k)-y2(m-k-1)<0&y2(m-k)<0.015
```

```

    lp = n-k;
    tp = (lp)*T;
    break
end
end
fprintf('%s%8.6f\n', ' Tiempo P = ',tp);
lx = round((lp+l)/2);
for n = lx:l
    if y2(n) > 0.45
        n;
        break
    end
end

% Calculo de la fase S
for k = lx:n+lx
    if (y2(n+lx-k)-y2(n+lx-k-1)<0)&y2(n+lx-
k)<0.09
        ls = n+lx-k;
        ts = (ls)*T;
        break
    end
end
fprintf('%s%8.6f\n', ' Tiempo S = ',ts);

% Plotear las gráficas
subplot(2,2,1), plot (t,y), grid on
title ('Grafica de la señal sismica')
fprintf('%s%8.6f\n', ' Amplitud = ',A);
Y = fft(y,N);
Pyy = Y.*conj(Y) / N;
f = (1/T)*(0:N/2)/ N;
subplot(2,2,3), plot(f,Pyy(1:N/2+1))
axis([0 1000 0 30]), grid on
title('Espectro de frecuencias de la señal')
xlabel('Frecuencia (Hz)')
N = 128; % N = numero de puntos TRF
for k = 1:N
    y1(k) = y(l-N/4+k);
end
subplot(2,2,2), plot (t(1:N),y1), grid on
title ('Zoom en el cambio de fase P-S')
Y = fft(y1,N);
Pyy = Y.*conj(Y) / N;
f = (1/T)*(0:N/2)/ N;
delta_f = f(2)-f(1);
subplot(2,2,4), plot(f,Pyy(1:N/2+1))
axis([0 1000 0 30]), grid on
title('Espectro de frecuencias del zoom')
xlabel('Frecuencia (Hz)')

% Calculo del periodo
[amp_espectro,J] = max (abs(Pyy));
fs = delta_f*(J-1);
periodo = 1/fs;
fprintf('%s%8.6f\n', ' Periodo = ',periodo);
% Calcular la duracion
prom_f = sum(abs(y(N-20:N)))/21;
for k = N:-1:l
    if y(k) > 4*prom_f
        k;
        tf = k*T+ss;
        break
    end
end
dur = round(tf-tp);
fprintf('%s%7.2f\n', ' Duracion = ',dur);

```

## References

- [1]. Udías, A., Mezcuca, J. Fundamentos de Geofísica, Primera Edición. Editorial Alhambra, Madrid (1996).
- [2]. Alan Oppenheim, Alan Willsky, Señales y Sistemas, Prentice Hall, México (1998).
- [3]. John Proakis, Dimitris Manolakis, Tratamiento digital de señales, Prentice Hall, Madrid (1998).
- [4]. Alejandro Nava, Procesamiento de series de tiempo, Fondo Cultura Económica, Mexico (2002).
- [5]. Herón Morales. Matlab 7, Métodos Numéricos, Grupo Editorial Megabyte, Lima (2005).
- [6]. V. Ingle & John Proakis, Digital Signal Processing using Matlab, PWS Publishing Company, (2004).
- [7]. Sydney Burrus & Alan Oppenheim. Ejercicios de Tratamiento de la Señal, utilizando Matlab V.4, Prentice Hall, Madrid (1998).