EJERCICIOS DE TRANSFORMADA DE FOURIER, TRANSFORMADA WAVELETS Y SUS RELACIONES

Por I. E. Vladimir Mosquera Cerquera, M. Sc. Unicauca, Univalle.

Objetivo: Material de soporte para que el estudiante profundice los conceptos y las relaciones existentes entre la transformada de Fourier, Transformada de Fourier por ventanas, transformada continua de Wavelets y transformada Discreta de Wavelets; todos los conceptos son explicados mediante ejemplos elementales de programación usando Matlab.

Considere las señales las cuales son generadas con una frecuencia de muestreo Fs de 40000 Hz.

$$s1(t) = 1sen(2\pi 1000t)$$

$$s(t) = s1 + 2sen(2\pi 2000t) + 3sen(2\pi 3000t)$$

$$s3(t) = \begin{cases} 0, & 0 \le t \le 0.0018 \text{ segundos} \\ 3sen(2\pi 3000t), & 0.0018 < t < 0.002 \text{ segundos} \end{cases}$$

$$ss(t) = s1(t) + 2sen(2\pi 2000t) + s3(t)$$

A continuación se muestran las señales.

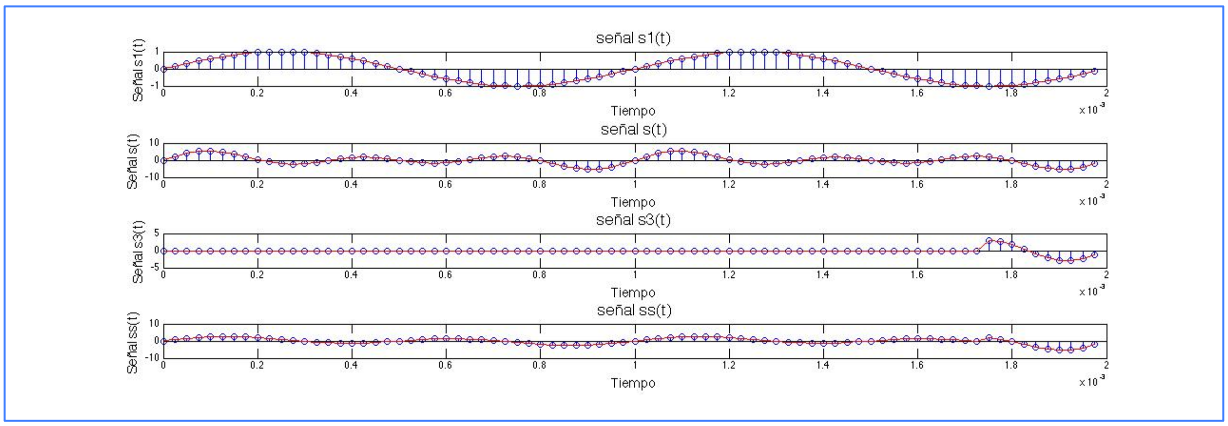

Figura 1. Señales en el dominio del tiempo a analizar.

1. ANALISIS DE FOURIER

Para entender la relación entre la transformada de Fourier por ventanas y la transformada de Fourier, considérese la función espectros_fourier(s, Fs, puntos, anchow, n_v) de la Figura 2. La línea 14 [B,F_e,t_e]=specgram(s,puntos,Fs,rectwin(anchow),0); usa la función specgram de la siguiente manera.

[B,f,t] = specgram(a,nfft,Fs,window,noverlap)

B es la matriz de salida, cada una de sus columnas corresponde al espectro de cada ventana de a calculado en las f frecuencias. B tendrá tantas columnas como tantas ventanas hayan y tendrá tantos renglones como tantas frecuencias sean definidas por el vector f.

f es el vector de frecuencias en las que es calculado el espectro, definidas por nfft y Fs. f=0: Fs/nfft:Fs/2.
t es el vector de tiempo donde cada elemento define el tiempo de inicio de cada ventana.
a es la señal a procesar.
nfft es la cantidad de puntos en los que es calculada la Transformada de Fourier.
window es el tipo de ventana usada.
noverlap es el traslape entre ventanas.

Para mayor información, investigue specgram en la ayuda de Matlab. Recomiendo investigar espectrogram, que es la versión actualizada de specgram; ambas funciones trabajan de manera muy parecida.

Un aspecto muy importante, es que la función espectros_fourier la hice con el objetivo de obtener la equivalencia entre la transformada de Fourier y la transformada de Fourier por ventanas *cuando las ventanas son rectangulares y cuando no hay traslape entre ellas.* Recomiendo investigar en la ayuda de Matlab como es usada specgram en la línea 14 de espectros_fourier.

```
1   function espectros_fourier(s, Fs, puntos, anchow, n_v)
2   % Esta función genera de marera gráfica el espectrograma de s, el espectro de magnitud de s
3   % obtenido del espectrograma y el espectro de la n_v-ésima ventana.
4   % s es la señal a procesar.
5   % puntos es el número de puntos de trnaformada de fourier.
6   % Fs es la Frecuencia de muestreo usada para obtener s.
7   % anchow es el ancho de la ventana a usar.
8   % n_v es la n_v-ésima ventana de s para obtener el espectro.
9   % Es recomendable que la longitud de s sea mútiplo entero de anchow. es
10  % decir, longitud de s=v*anchow; v es el número de ventanas.
11  % Función hecha por Vladimir Mosquera.
12  F=0:Fs/length(s):Fs-Fs/length(s);
13  figure; subplot(3,1,1); specgram(s,puntos,Fs,rectwin(anchow),0);
14  [B,F_e,t_e] =specgram(s,puntos,Fs,rectwin(anchow),0); title('Espectrograma');
15  subplot(3,1,2);
16  if puntos==anchow
17      stem(F_e, abs(B));
18      title('Espectro de Magnitud Obtenido del Espectrograma de la Señal s(t) Usando Una Ventana');
19      xlabel('Frecuencia en Hz'); ylabel('abs(B)');
20  else stem3(t_e, F_e, abs(B));
21        v=round(puntos/anchow);
22        title(['Espectro de Magnitud Obtenido del Espectrograma de la Señal s(t) Usando' ' ' num2str(v) ' ' 'Ventanas']);
23        xlabel('Tiempo'); ylabel('Frecuencia en Hz'); zlabel('abs(B)');
24  end;
25  mag_S1v=abs(fft(s(n_v*anchow-anchow+1:n_v*anchow), puntos));
26  subplot(3,1,3); stem(F, mag_S1v);
27  title(['Espectro de Magnitud Obtenido del Espectrograma de la señal s(t):' ' ' num2str(n_v) ' ' 'Ventana']);
28  xlabel('Frecuencia en Hz'); ylabel('mag_S1v');
```

Figura 2. Función espectros_fourier.

Al ejecutar >> espectros_fourier(s, 40000, 80,80, 1), se obtienen los resultados de la Figura 3. Observe como el ancho de la ventana es igual al numero de instantes en los que esta definida la señal s (80 puntos), implicando que se ha obtenido la transformada de Fourier por ventanas *en una sola ventana*. Por esta razón, la transformada de Fourier por ventanas en una sola ventana calculada por especgram es equivalente a la transformada de Fourier calculada por fft, evidenciándose esto en las dos ultimas gráficas de la Figura 3. La única diferencia es que especgram calcula el espectro hasta Fs/2; fft lo calcula hasta Fs.

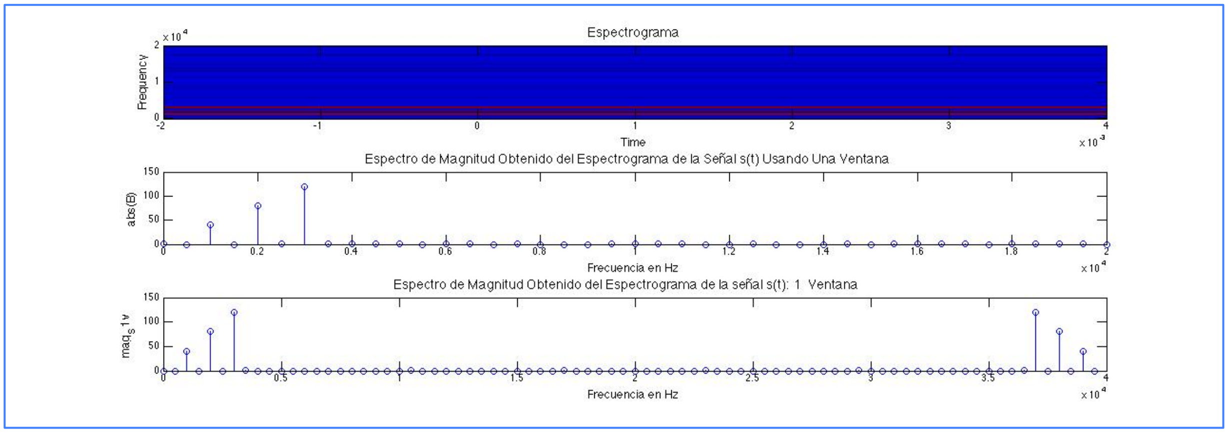

Figura 3. Resultados obtenidos por la función espectros_fourier para $s(t)$.

Al ejecutar >> espectros_fourier(ss, 40000, 80,20, 4), se obtienen los resultados de la Figura 4.

Observe como el ancho de la ventana es igual a 20 puntos, implicando que se ha obtenido la transformada de Fourier por ventanas *en cuatro ventanas*. Por esta razón, la transformada de Fourier por ventanas en la cuarta ventana calculada por especgram es equivalente a la transformada de Fourier calculada por fft en los últimos 20 puntos de $ss(t)$ o últimos 20*1/40000=5e-4 segundos, evidenciándose esto en las dos ultimas gráficas de la Figura 4. Observe como el espectro de la cuarta ventana de la gráfica dos es equivalente al espectro de la gráfica 3. Como antes, la única diferencia es que especgram calcula el espectro hasta Fs/2; fft lo calcula hasta Fs.

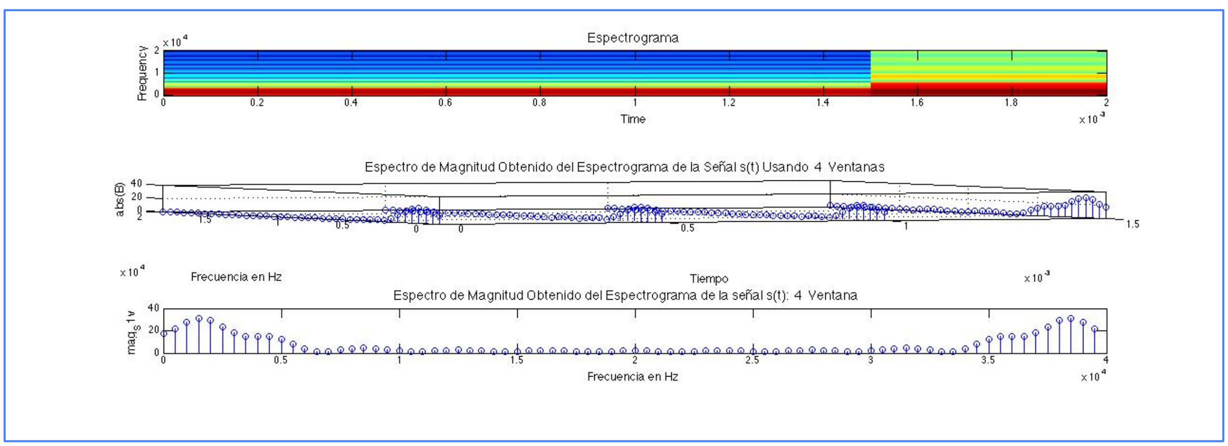

Figura 4. Resultados obtenidos por la función espectros_fourier para $ss(t)$.

En este experimento, la variable t_e da como resultado

```
t_e =
        0
   0.0005
   0.0010
```

lo cual implica que la cuarta ventana inicia en 0.0015 segundos. Además, sabiendo que la contribución de $s3(t)$ es en los últimos 2.5e-4 segundos de $ss(t)$, en este experimento concluye que el análisis de Fourier por ventanas genera una incertidumbre de 5e-4 segundos (el ancho de la ventana en tiempo) en la contribución de $s3(t)$ en $ss(t)$.

Ejercicio: hacer el experimento para $s1(t)$ y $s3(t)$ y obtener buenas conclusiones.

2. ANALISIS CONTINUO DE WAVELETS

Para interpretar los coeficientes de Wavelets, considérese el script mostrado en la Figura 5.

```
1 -   x = zeros(100,1);
2 -   x(1) = 1; x(50) = 1; x(100) = 1;
3 -   subplot (4,1,1);  stem(x); hold on; plot(x, 'r'); hold off; title('Señal de Implusos'); xlabel('Espacio'); ylabel('x');
4 -   CWTcoeffs = cwt(x,1:50,'haar');
5 -   subplot (4,1,2);  stem(CWTcoeffs(10,:)); hold on; plot(CWTcoeffs(10,:), 'r'); hold off;
6 -   title('Coeficientes Escala 10'); xlabel('Espacio'); ylabel('Ca=10');
7 -   subplot (4,1,3);  stem(CWTcoeffs(30,:)); hold on; plot(CWTcoeffs(30,:), 'r'); hold off;
8 -   title('Coeficientes Escala 30'); xlabel('Espacio'); ylabel('Ca=30');
9 -   subplot (4,1,4);  stem(CWTcoeffs(50,:)); hold on; plot(CWTcoeffs(50,:), 'r'); hold off;
10 -  title('Coeficientes Escala 50'); xlabel('Espacio'); ylabel('Ca=50');
```

Figura 5. Script para el análisis Wavelets una señal de impulsos usando funciones Haar.

Al ejecutar el script se obtienen los resultados de la Figura 6.

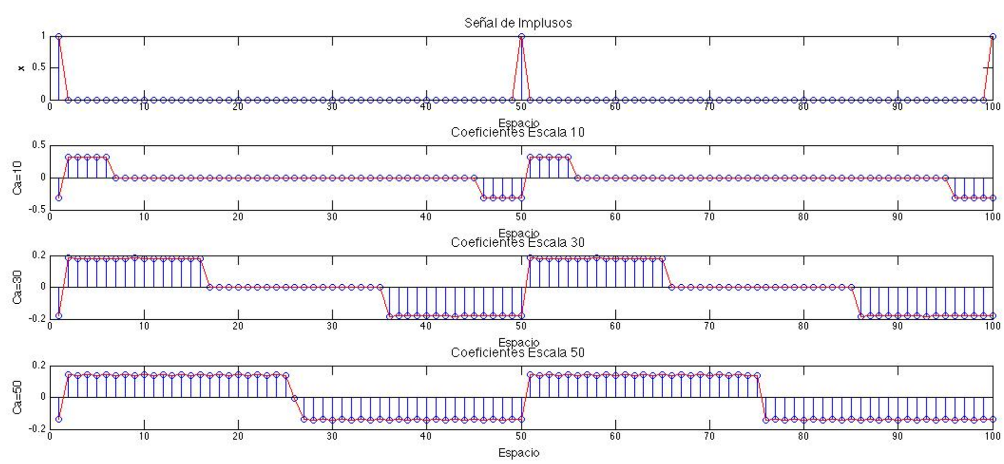

Figura 6. Análisis Wavelets a una señal de impulsos en las escalas 10, 30 y 50.

La señal a analizar son tres impulsos ubicados en la posición uno, cincuenta y cien de un vector de cien elementos. La línea 4 usa la función cwt, donde CWTcoeffs es una matriz donde cada m-ésimo renglón corresponde a la escala m en la que es calculada la transformada continua de Wavelets. Cada n-ésima columna corresponde a un punto de tiempo o espacio en la que es calculada la transformada continua de Wavelets, que en analogía con las ventanas de Fourier, corresponden a ventanas del menor ancho posible de una muestra.

Observe cómo al operar la función Haar con cada impulso de x, da como resultado la misma función Haar reflejada en su línea de simetría vertical; este resultado es debido a la "convolución" de la función Haar a medida que va pasando por cada impulso de x.

Cada escala corresponde a los puntos en los que esta definida la función Haar y por consiguiente su duración espacial o temporal. En este experimento, por ejemplo para la escala 30, la función Haar tiene una duración espacial de 30 muestras o de 30/Fs segundos, donde Fs es la frecuencia de muestreo en Hz.

Ejercicio: hacer el experimento para otras familias de Wavelets y obtener buenas conclusiones.

Al ejecutar

```
figure;
cwt(s1,1:80,'haar','3Dplot');
colormap jet; colorbar;
figure;
subplot(3,1,1); cwt(s1,1:80,'haar','plot');
colormap jet;
CWTcoeffs = cwt(s1,1:80,'haar');
subplot(3,1,2); plot(t,s1,'r'); grid on; title('seÒal s1(t)'); xlabel('Tiempo'); ylabel('SeÒal s1(t)');
x = zeros(80,1);
x(41) = 1;
CWTcoeffsx = cwt(x,1:80,'haar');
subplot(3,1,3); plot(CWTcoeffsx(40,:),'r'); grid on; title('Haar escala 40'); xlabel('Espacio');
ylabel('Haar escala 40');
```

se obtienen los resultados mostrados en las Figuras 7 y 8.

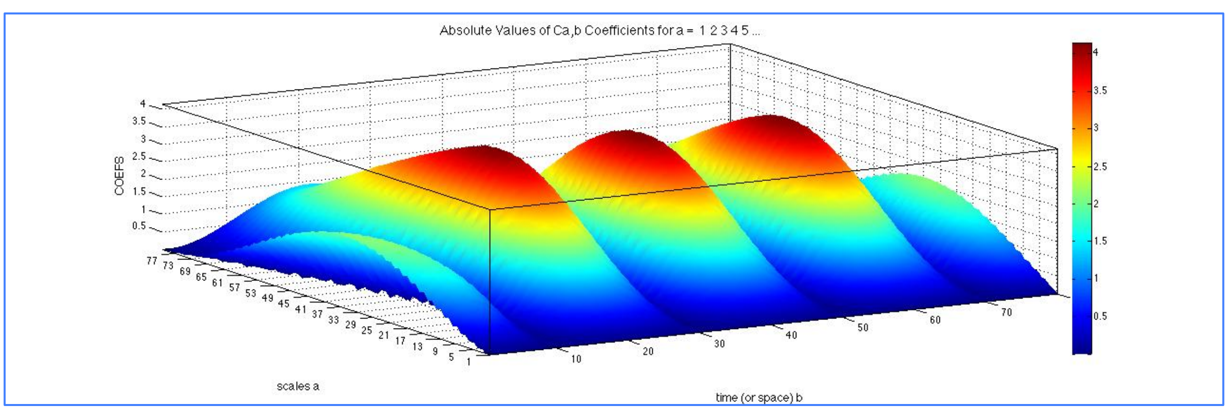

Figura 7. Espectrograma 3D usando la transformada continua de Wavelets para la señal $s1(t)$.

Observe como en la Figura 7, los coeficientes mas calientes (en color rojo) corresponden en tiempo a los cruces por cero de $s1(t)$ y en escala aproximadamente a mitad de escala (80/2)=40. La Figura 8 aclara mas este concepto.

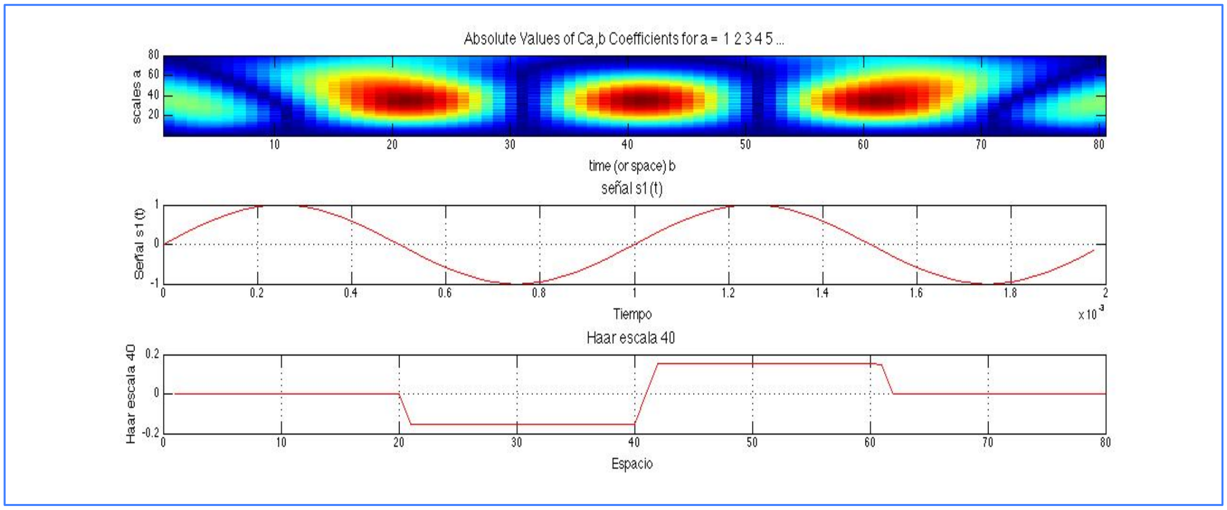

Figura 8. Espectrograma usando la transformada continua de Wavelets para la señal $s1(t)$ con escala 40 y espacio 41 para localizar el cruce por cero $s1(t)$ de en 0.001 segundos.

De la Figura 8 se observa que los coeficientes presentan más alto valor para los cruces por cero de la señal $s1(t)$. Para el tiempo de 0.0001 segundos se analiza que el valor alto en los coeficientes se obtiene para la escala 40, cuya Haar tiene una duración de aproximadamente el periodo de $s1(t)$. Esto se aplica en los tiempos donde hay cruce por cero en $s1(t)$.

Al ejecutar el código

```
Fs=40000;
t=0:1/Fs:0.002-1/Fs;
x=sin(2*pi*1000*t)+sin(2*pi*2000*t)+sin(2*pi*3000*t);
cwt(x,1:80,'haar','3Dplot');
colormap jet; colorbar;
```

se obtiene el resultado de la Figura 9. Que puede concluir?

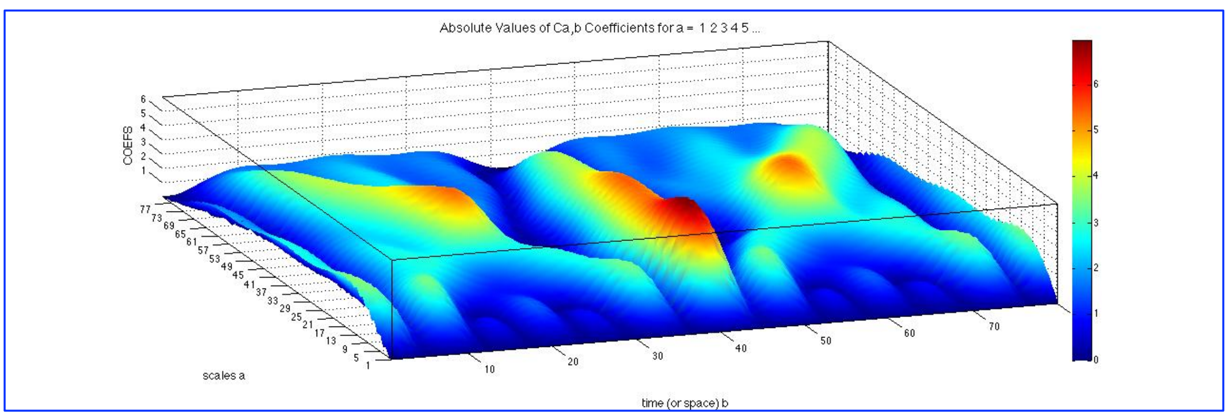

Figura 9. Espectrograma 3D Wavelets para la señal $x(t)$.

Ejercicio. Obtener el espectrograma 3D Wavelets para las señales $s(t)$, $s3(t)$ y $ss(t)$. Hacer buenas conclusiones.

3. ANALISIS DISCRETO DE WAVELETS

Para interpretar los coeficientes de la transformada discreta de Wavelets, considérese el código mostrado a continuación.

```
Fs=40000;
t=0:1/Fs:0.002-1/Fs;
s=1*sin(2*pi*1000*t)+2*sin(2*pi*2000*t)+3*sin(2*pi*3000*t);
figure; stem(t,s);hold on; plot(t,s,'r'); hold off; title('SeÒal s(t)'); xlabel('Tiempo'); ylabel('Amplitud');
[c,l]=wavedec(s,3,'sym2'); % s es s(t), la seÒal a analizar. La descomposiciÛn se hace hasta el nivel 3.
cA3=c(1:l(1));
cD3=c(l(1)+1:l(1)+l(2));
figure; Nivel=3; t3=(0:(length(cD3)-1)).*(2^Nivel/Fs); % Se calcula el tiempo para este nivel de
resoluciÛn.
subplot(3,1,1); stem(t3,cA3); hold on; plot(t3,cA3,'r');hold off; title('cA3'); xlabel('Tiempo');
ylabel('Valor');
subplot(3,1,2); stem(t3,cD3); hold on; plot(t3,cD3,'r');hold off; title('cD3'); xlabel('Tiempo');
ylabel('Valor');
subplot(3,1,3);stem(t3,cA3+cD3); hold on; plot(t3,cA3+cD3,'r');hold off; title('cA3+cD3');
xlabel('Tiempo'); ylabel('Valor');
cD2=detcoef(c,l,2);           % Se extrae los coeficientes de detalle cD2. (Segundo nivel).
cA2=appcoef(c,l,'sym2',2);    % Se extrae los coeficientes de escala cA2. (Segundo nivel)
                              % usando una Wavelet sym2.
figure; Nivel=2; t2=(0:(length(cD2)-1)).*(2^Nivel/Fs);  % Se calcula el tiempo para este nivel de
resoluciÛn.
subplot(3,1,1); stem(t2,cA2); hold on; plot(t2,cA2,'r');hold off; title('cA2'); xlabel('Tiempo');
ylabel('Valor');
subplot(3,1,2); stem(t2,cD2); hold on; plot(t2,cD2,'r');hold off; title('cD2'); xlabel('Tiempo');
ylabel('Valor');
subplot(3,1,3);stem(t2,cA2+cD2); hold on; plot(t2,cA2+cD2,'r');hold off; title('cA2+cD2');
xlabel('Tiempo'); ylabel('Valor');
```

La función wavedec hace la descomposición Wavelets como ya se explico en clase, pero es importante que la estudie en el help de Matlab. Las funciones detcoef y appcoef operan de la siguiente manera.

D = detcoef(C,L,N)

Extrae los coeficientes de detalle Wavelets en el nivel N a partir de los vectores C y L obtenidos por la función wavedec. Por ejemplo, si N=1, en D se extraen los últimos L(end-1) elementos de C.

A = appcoef(C,L,'wname',N)

Funciona muy parecida a la función anterior. wname es la función Wavelets a usar. Por ejemplo, si N=1 y wname=haar, en A se extraen los coeficientes cA1 que han sido calculados a partir de la función Wavelets Haar.

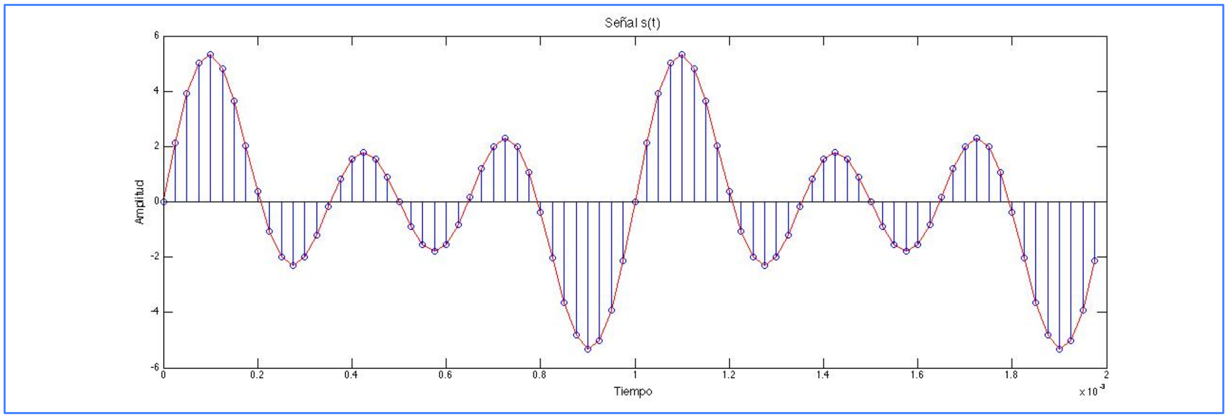

Figura 10. Señal $s(t)$.

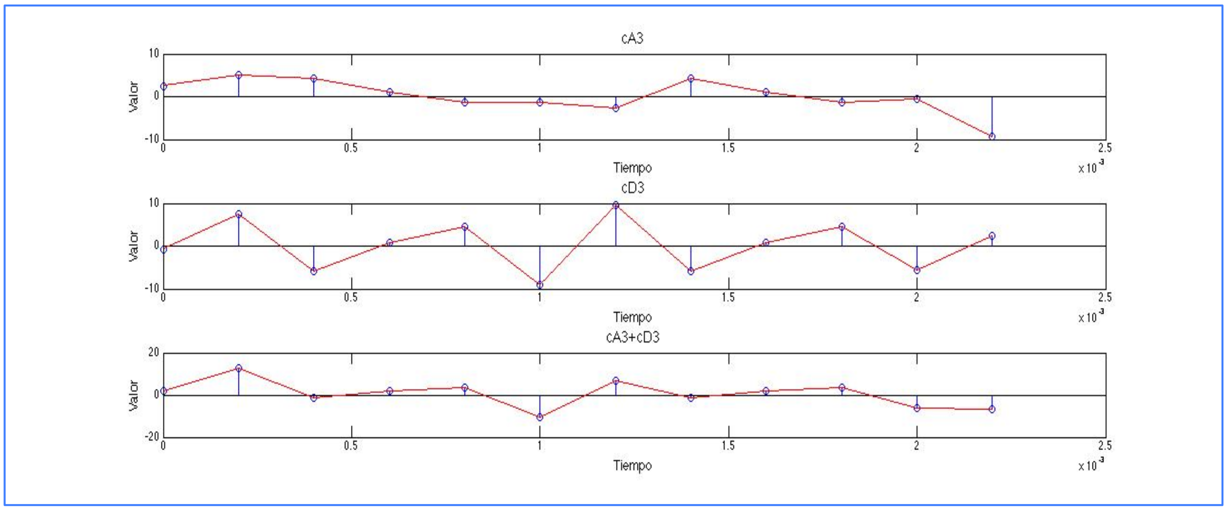

Figura 11. Descomposición de $s(t)$ en el Nivel 3.

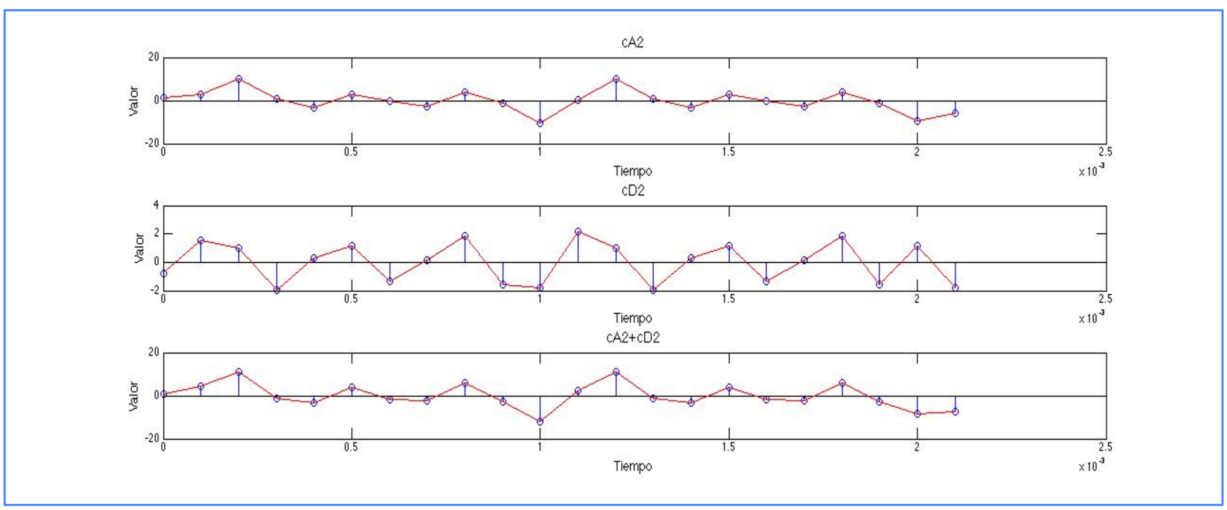

Figura 12. Descomposición de $s(t)$ en el Nivel 2.

En la Figura 11 se observa la descomposición en el nivel 3. Observe como $s(t)$ ha sido submuestreada en un factor de 2^3 para la obtención de cA3 y cD3; por esta razón la separación en las muestras temporales de cA3 y cD3 es de 8/Fs segundos.

Conclusiones similares se pueden hacer de la Figura 12 al usar descomposición en el nivel 2.

¿Que puede concluir si se ejecuta la instrucción `cD2_=c(25:24+l(3));` ?