
Guia 3 - Ejercicio 4

Table of Contents

Enunciado	1
Inciso a	1
Inciso b	2
Inciso c	2
Inciso d	2
Inciso e	5
Anexo	8

Enunciado

Sobre la misma señal, explore la descomposición ondita generada mediante la transformada ondita diádica discreta (también llamada DWT), que es el algoritmo explorado en la sección 7.3.1:

a) La función `dwt` permite generar un nivel de descomposición, es decir, a partir de la señal en un espacio, obtiene los coeficientes de aproximación y detalle de la siguiente escala. Explore su uso, generando iterativamente la % descomposición hasta nivel $L = 3$.

b) La función `idwt` permite reconstruir un nivel de la descomposición. Explore la reconstrucción para la señal descompuesta hasta escala $L = 3$ del caso anterior.

c) Compare sus resultados con las funciones `wavedec` y `waverec`, que permiten realizar la descomposición y reconstrucción completa hasta el nivel deseado.

d) Escriba una función en Matlab que permita visualizar en graficas separadas, la señal original y los coeficientes de la descomposición para cada escala. Por ejemplo, si se aplicó una descomposición hasta nivel 2, debe producir una gráfica con 4 subgráficas, una para la señal original, otra para los coeficientes de detalle para la escala $j = 1$, otra para los coeficientes de detalle en escala $j = 2$ y otra para los coeficientes de aproximación a escala $j = 2$. Pruebe esta función para la señal seleccionada, haciendo una descomposición hasta el nivel 3.

e) A partir de la descomposición obtenida con `wavedec`, la función `wrcoef` permite obtener la reconstrucción a partir de los coeficientes de aproximación de detalle del nivel que se desee, es decir, permite obtener la proyección sobre V_j y W_j . Escriba una función que permita graficar en subgráficas, la señal original y las proyecciones en los subespacios W_1, \dots, W_p y V_p para una descomposición con p niveles. Pruebe esta función sobre una descomposición multiresolución hasta nivel $p = 3$.

```
clear all;
close all;
```

```
load internet-traffic-data-in-bits-fr1
```

```
x = internet_traffic_data_in_bits_ft_quantity(512:512+2^11-1);
```

Inciso a

```
wav = 'db9';
```

```
[cA1,cD1] = dwt(x,wav,'per');
[cA2,cD2] = dwt(cA1,wav,'per');
[cA3,cD3] = dwt(cA2,wav,'per');
```

Inciso b

```
xr2 = idwt(cA3,cD3,wav,'per');
fprintf('La reconstruccion de nivel 3 tiene un error de %.4f\n', ...
        norm(cA2-xr2));
xr1 = idwt(xr2,cD2,wav,'per');
fprintf('La reconstruccion de nivel 1 tiene un error de %.4f\n', ...
        norm(cA1-xr1));
xr0 = idwt(xr1,cD1,wav,'per');
fprintf('La reconstruccion de nivel 0 tiene un error de %.4f\n', ...
        norm(x-xr0));
```

```
La reconstruccion de nivel 3 tiene un error de 0.2282
La reconstruccion de nivel 1 tiene un error de 0.3655
La reconstruccion de nivel 0 tiene un error de 0.4569
```

Inciso c

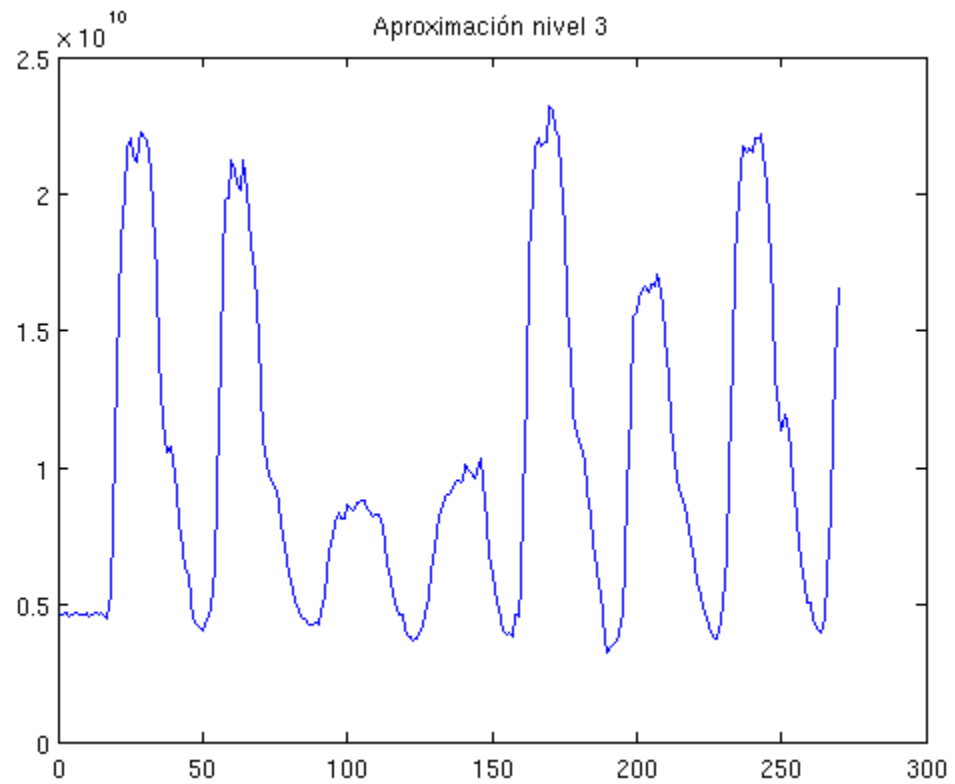
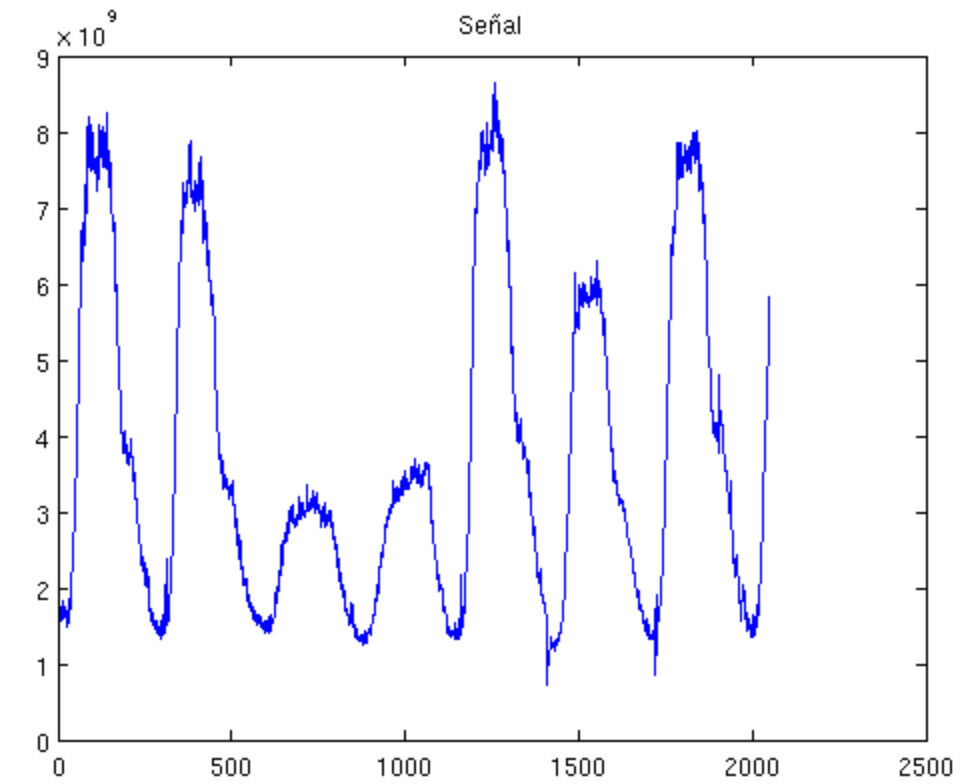
```
[C,L] = wavedec(x,3,wav,'per');
c1_err = norm(C-[cA3 cD3 cD2 cD1]);
fprintf('La descomposición tiene un error de %.4f\n', ...
        c1_err);

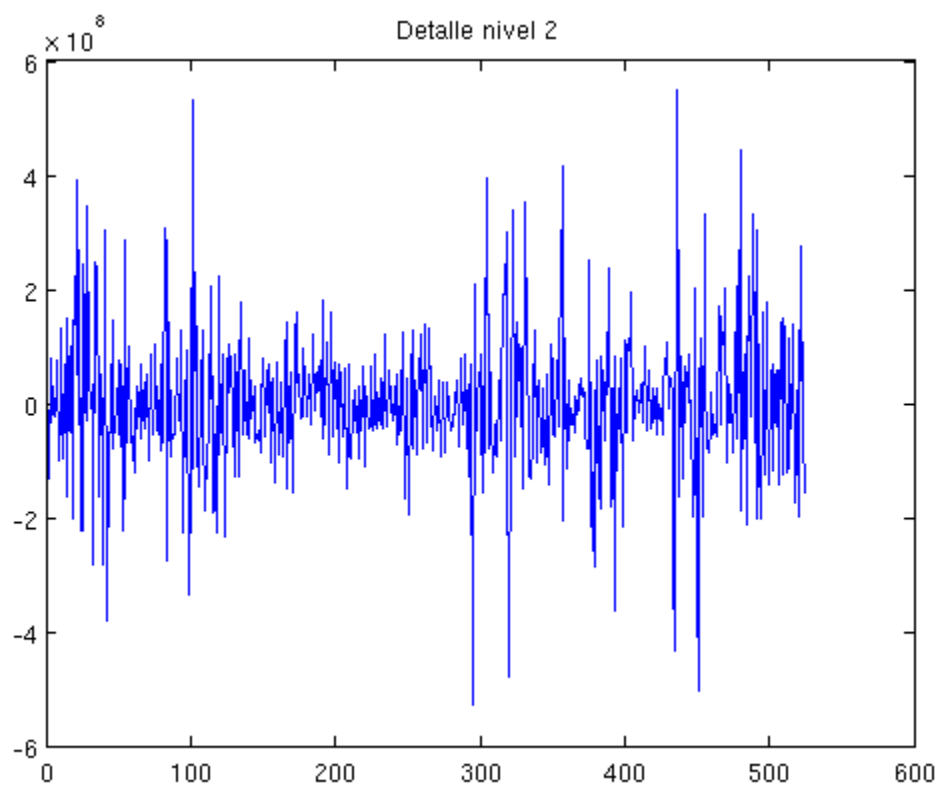
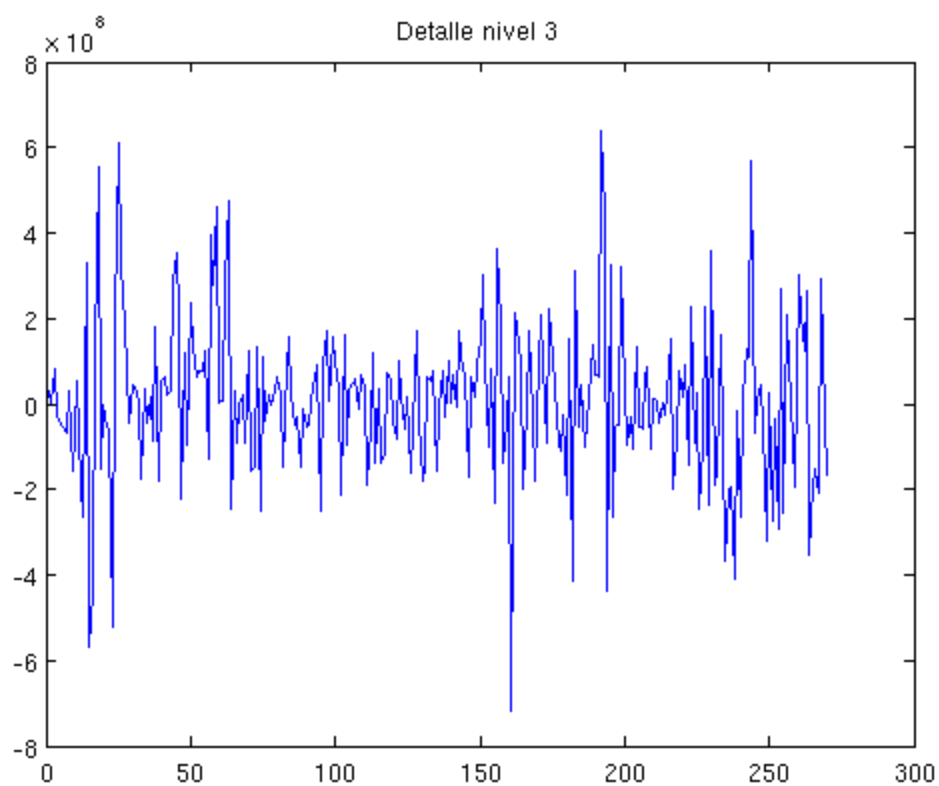
xrec_c = waverec(C,L,wav);
c2_err = norm(xrec_c-xr0);
fprintf('La reconstruccion tiene un error de %.4f\n', ...
        c2_err);
```

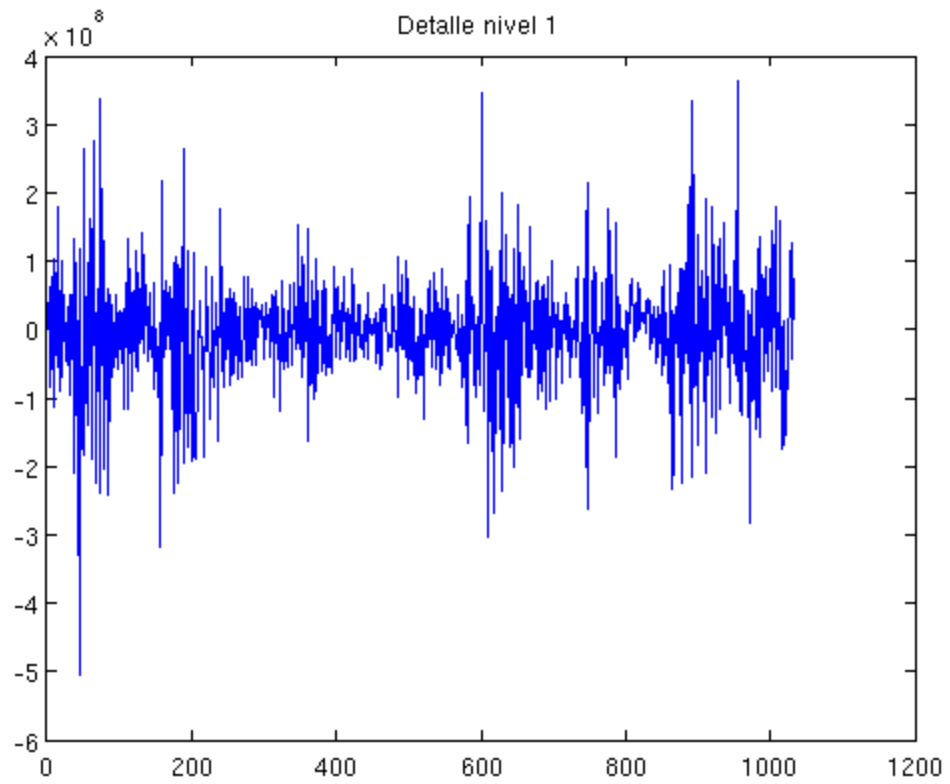
```
La descomposición tiene un error de 0.0000
La reconstruccion tiene un error de 0.0000
```

Inciso d

```
graf_wavedec(x,C,L)
```

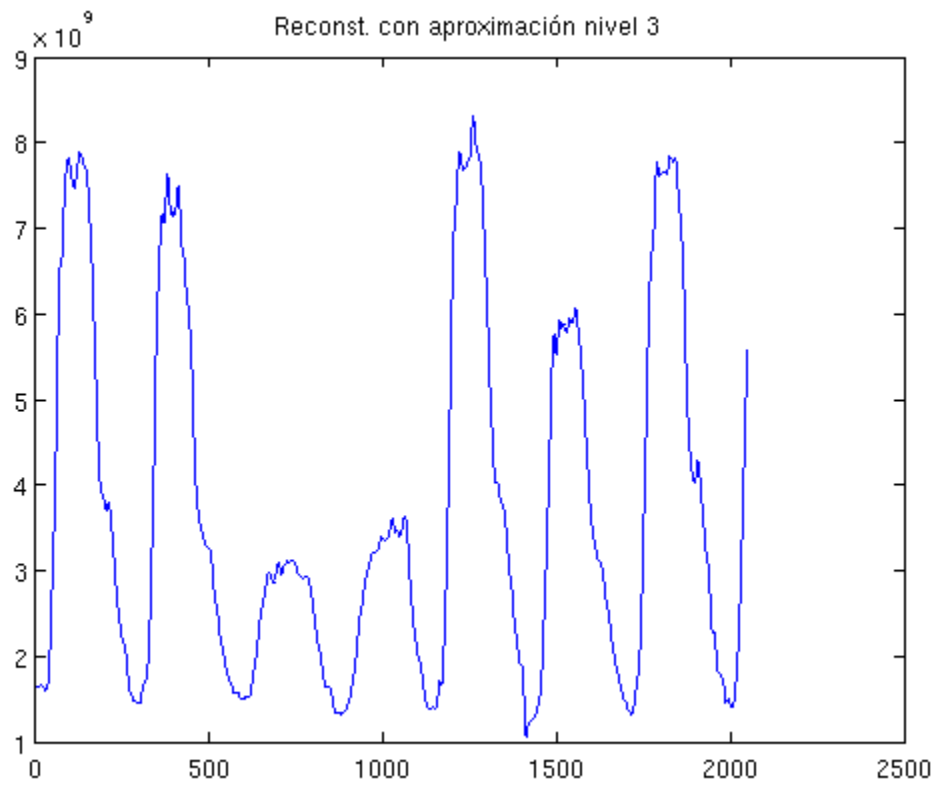
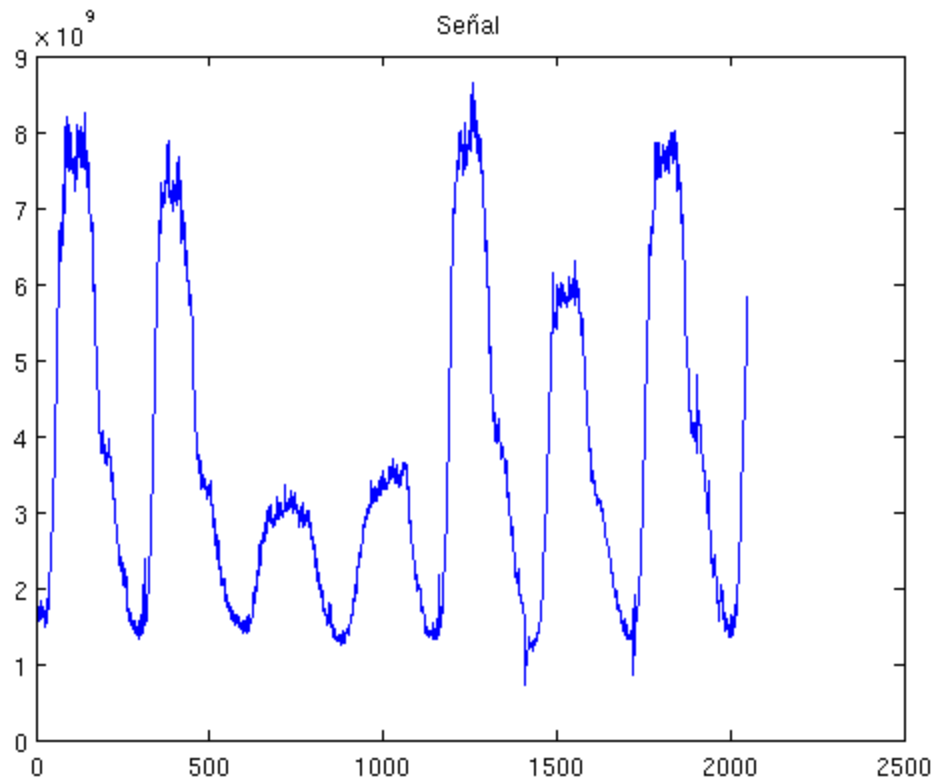


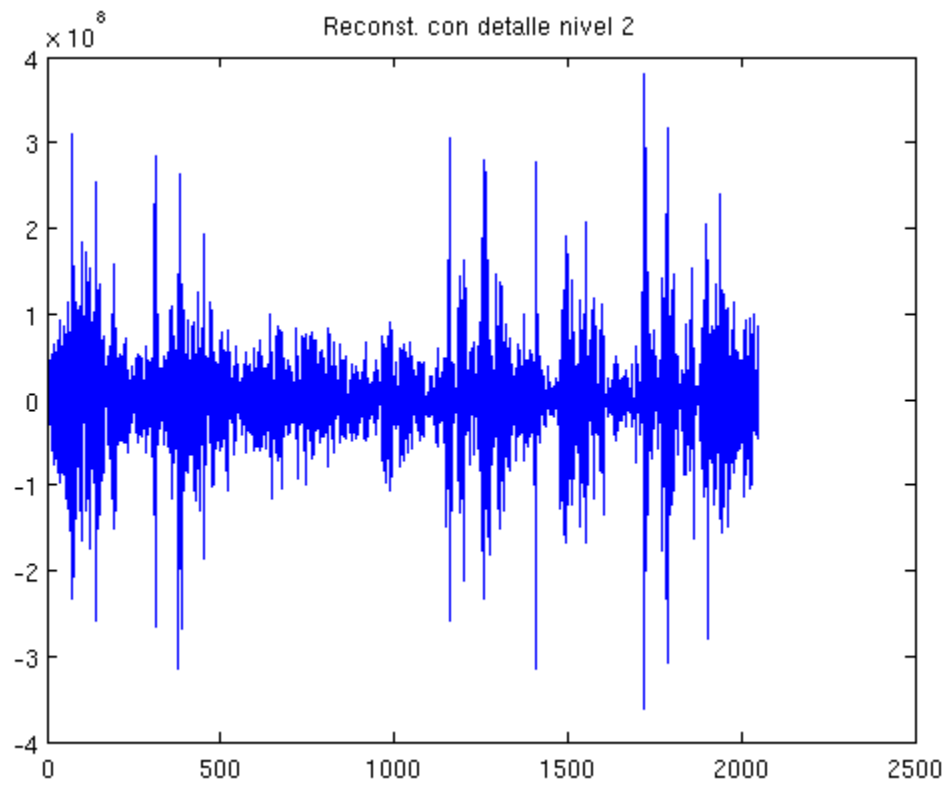
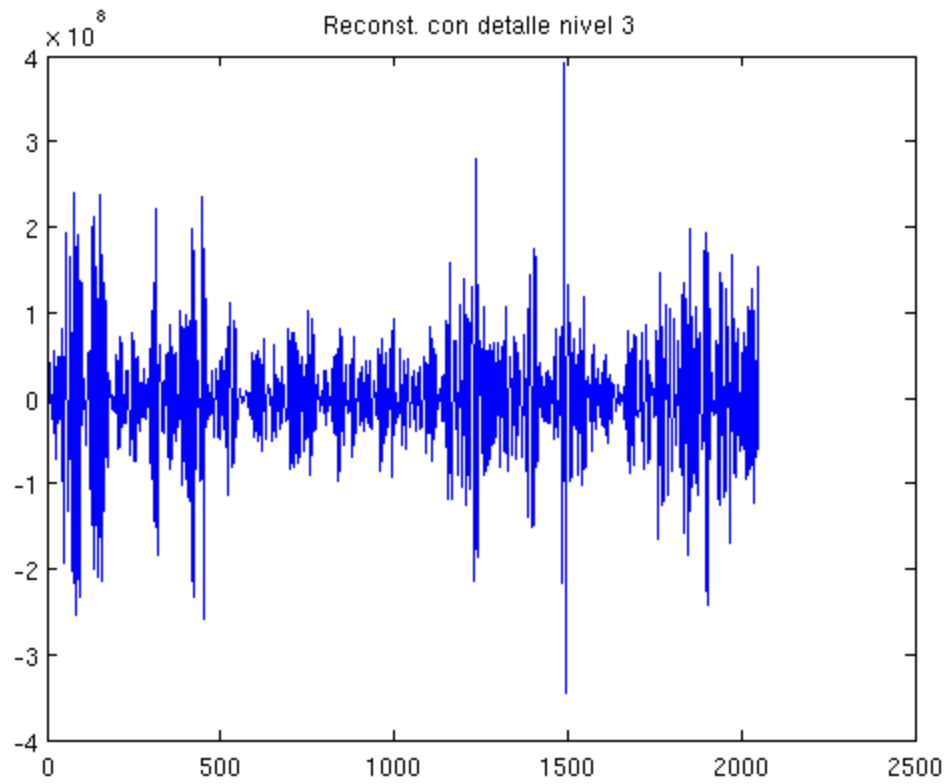


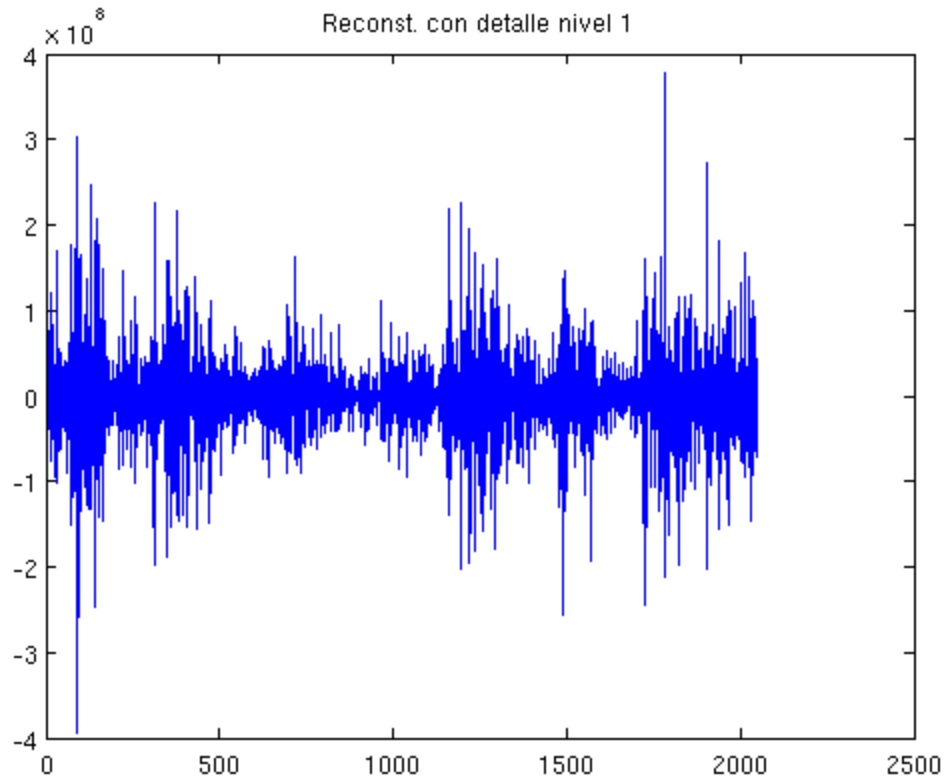


Inciso e

`graf_wrcoef(x,C,L,wav)`







Anexo

Función graf_wavedec

dbtype graf_wavedec.m

```

1    function graf_wavedec( x, coef, largo )
2
3    NMAX = length(largo)-2;
4    figure()
5    plot(x);
6    title('Señal');
7    for n=1:1:NMAX+1
8        if n==1
9            figure()
10           plot(coef(1:largo(1)));
11           title(sprintf('Aproximación nivel %d',NMAX-(n-1)));
12        else
13            figure()
14            p1 = sum(largo(1:n-1))+1;
15            p2 = sum(largo(1:n));
16            plot(coef(p1:p2));
17            title(sprintf('Detalle nivel %d',NMAX-(n-2)));
18        end
19

```



```
20     end
21
```

Función graf_wrcoef

dbtype graf_wrcoef.m

```
1     function graf_wrcoef( x, coef, largo, wav )
2
3     NMAX = length(largo)-2;
4     figure()
5     plot(x);
6     title('Señal');
7     for n=1:1:NMAX
8         if n==1
9             figure()
10            plot(wrcoef('a',coef,largo,wav,NMAX-(n-1)));
11            title(sprintf('Reconst. con aproximación nivel %d', ...
12                NMAX-(n-1)));
13            figure()
14            plot(wrcoef('d',coef,largo,wav,NMAX-(n-1)));
15            title(sprintf('Reconst. con detalle nivel %d', ...
16                NMAX-(n-1)));
17        else
18            figure()
19            plot(wrcoef('d',coef,largo,wav,NMAX-(n-1)));
20            title(sprintf('Reconst. con detalle nivel %d', ...
21                NMAX-(n-1)));
22        end
23    end
24 end
25
```

Published with MATLAB® R2013a