

Tratamiento de señales. Periodo de edificaciones

- Nombre: Ing. Patricio Palacios Benavides
- Fecha: Mayo 2020
- Línea de Investigación: Estructuras
- Proyecto:

Tratamiento de señales. Periodo de edificaciones

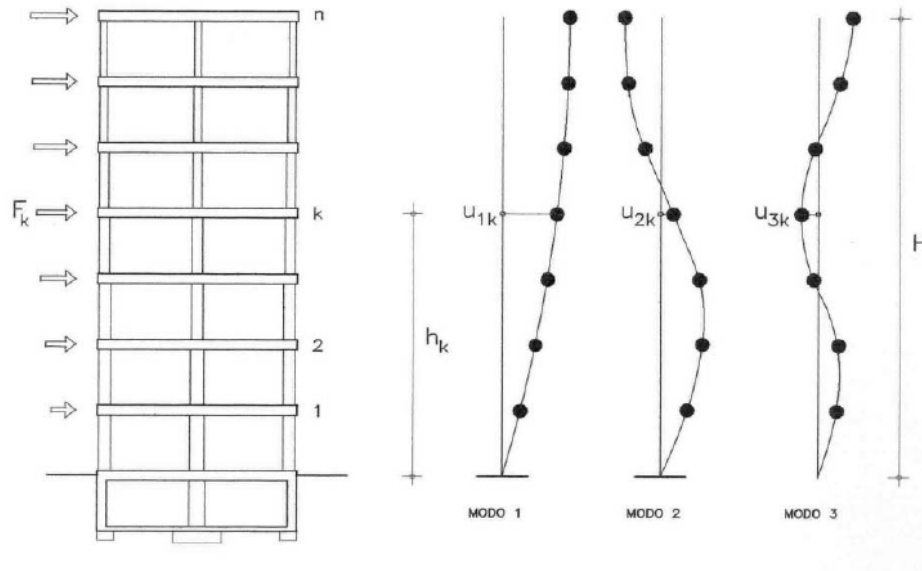
Patricio Palacios B.

Ingeniero Civil Mención Estructuras



El objetivo es implementar las herramientas necesarias mediante algoritmos para la caracterización de las estructuras regulares de hormigón armado mediante el periodo de vibración del primer modo. Se realizarán algoritmos avanzados de procesamiento digital de señales. Como herramientas matemáticas se utilizará la transformada discreta de Fourier, filtros de señales y algoritmos de la aplicación **Signal Analyzer** de Matlab.

El propósito de los resultados es calibrar los modelos matemáticos, mediante la variación de los parámetros mecánicos utilizados, con el fin de generar respuestas similares a las de la edificación real.



El procesamiento que se realizará sobre las señales registradas hasta la transformación en un conjunto de vectores de representación de características, se divide en dos partes:

1. Pre-procesamiento de datos: indica el filtrado y normalización de las señales sísmicas antes de la caracterización.

- Filtrado

- Normalización

2. Extracción de características: transformación de las muestras de las señales sísmicas a vectores de características.

- Procesamiento basado en el tiempo
- Procesamiento basado en la frecuencia



Patricio Xavier Palacios Benavides

Ingeniero Civil Mención Estructuras

Table of Contents

1. Simbología.....	2
2. Lectura de archivo con datos.....	3
2.1 Propiedades de la señal.....	3
3. Algoritmo de activacion STA/LTA.....	3
4. Algoritmo de seleccion de ventanas.....	5
5. Algoritmo de aplicación de filtro.....	5
6. Algoritmo para Transformada Rápida de Fourier.....	7
7. Algoritmo para suavizar (Konno & Ohmachi 1998)	8
8. Algoritmo para Promedio.....	9
9. Gráfica de Ventanas.....	9
9.1 Periodo predominante en ventanas.....	9
9.2 Espectro de potencia.....	12
9.3 Escalograma de magnitud.....	13
10. Amplificación dinámica.....	14
9.1 Vibración Armónica Amortiguada.....	14
Referencias Bibliograficas.....	16

```
cd 'D:\OneDrive\01. Brain\01. Consultorias\048. Reforzamiento San
Gabriel\06. SCRIPT\06. AmbientalVibration';
% cd 'D:\OneDrive\01. Brain\01. Consultorias\60. Quitumbe UTOPIA\06.
CálculoEstructural\AmbientalVibration';
clear;clc;close all;
```

1. Simbología

```
% Fs      = Frecuencia de muestreo
% T        = Tiempos registrados
% V        = Variable de señales
% STA      = Tiempo STA (1s) - Short Term Average
% LTA      = Tiempo LTA (30s) - Long Term Average
```

```
% vent = Tiempo de cada ventana (25s)
% vmin = Valor mínimo STA/LTA
% vmax = Valor máximo STA/LTA
% p = Porcentaje de Tapper (5%)
% f1 = Frecuencia 1 para filtro (1 Hz)
% f2 = Frecuencia 2 para filtro (4 Hz)
% bexp = Constante para suavizado (40-100)
```

2. Lectura de archivo con datos

```
load Suelo01_01.txt;
Vo=Suelo01_01(:,1);
Testr=0.2; % Periodo estructural
```

2.1 Propiedades de la señal

```
% Frecuencia de muestreo
Fs=244;
% Duracion de las ventanas
vent=25;
% Algoritmo STA/LTA
STA=1; LTA=30;
% Limites algoritmo STA/LTA
vmin=0.5; vmax=1.5;
% R para tapper de ventana
p=0.05;
% Límites de ancho de banda de frecuencias
f1=1; f2=50;
% Constante de suavizado (Konno & Ohmachi 1998 )
bexp=40;
```

3. Algoritmo de activacion STA/LTA

El algoritmo más sencillo para la detección de un evento sísmico se basa en comparar el nivel de la señal respecto a un umbral de disparo (Pérez, Cuartas, & Ferrer, 2006). Sin embargo, las fluctuaciones de alta frecuencia que usualmente se asocian al ruido, hacen que se generen falsos disparos, pues con niveles altos de ruido sísmico se están disparando continuamente (Peña, 2014).

Una solución para eliminar dichas fluctuaciones y que surja un contenido con información relevante, es utilizar el algoritmo denominado STA/LTA que se puede traducir como razón promedio de corto plazo versus largo plazo (Peña, 2014), donde STA es la amplitud media en un intervalo temporal corto (t_{STA}) entre 1 a 4 segundos y LTA es la amplitud media en un intervalo temporal largo (t_{LTA}) entre 8 a 32 segundos (Pérez, Cuartas, & Ferrer, 2006).

$$N_s = F_s * t_{STA}$$

$$N_T = F_s * t_{LTA}$$

$$STA = \frac{1}{N_s} \sum_j^i = i - N_s * x_j^2$$

$$LTA = \frac{1}{N_T} \sum_j^i = i - N_T * x_j^2$$

$$j \leq 0$$

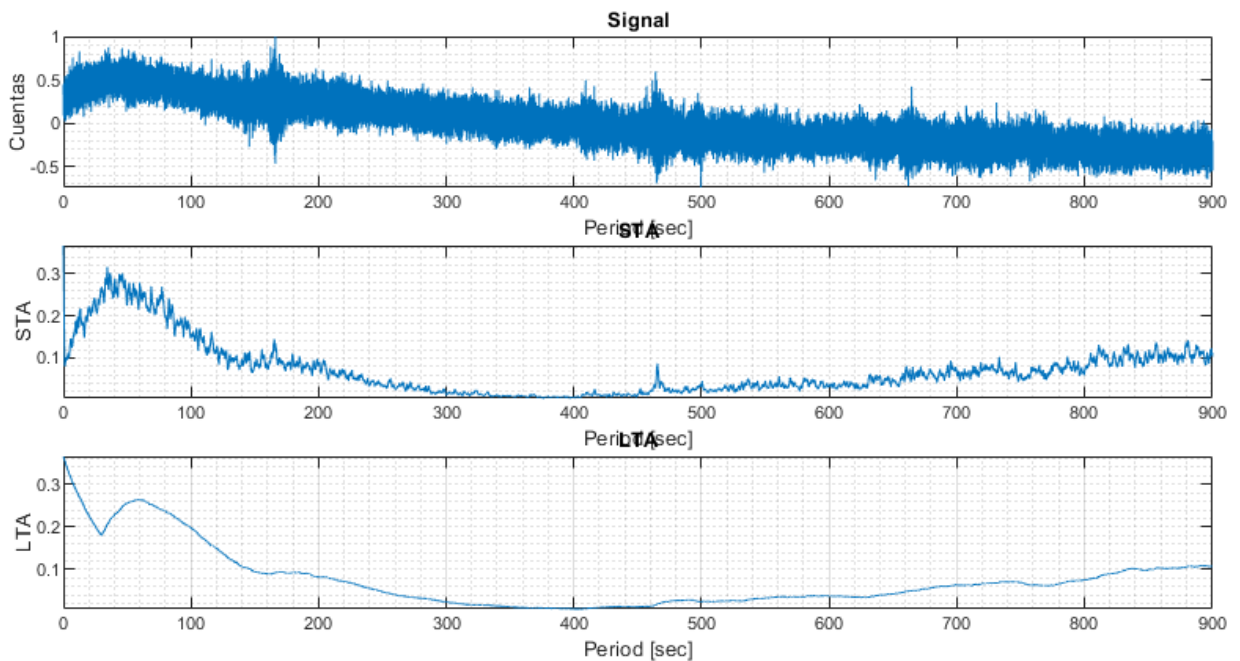
$$x_j = \frac{(x_1 + x_2)}{2}$$

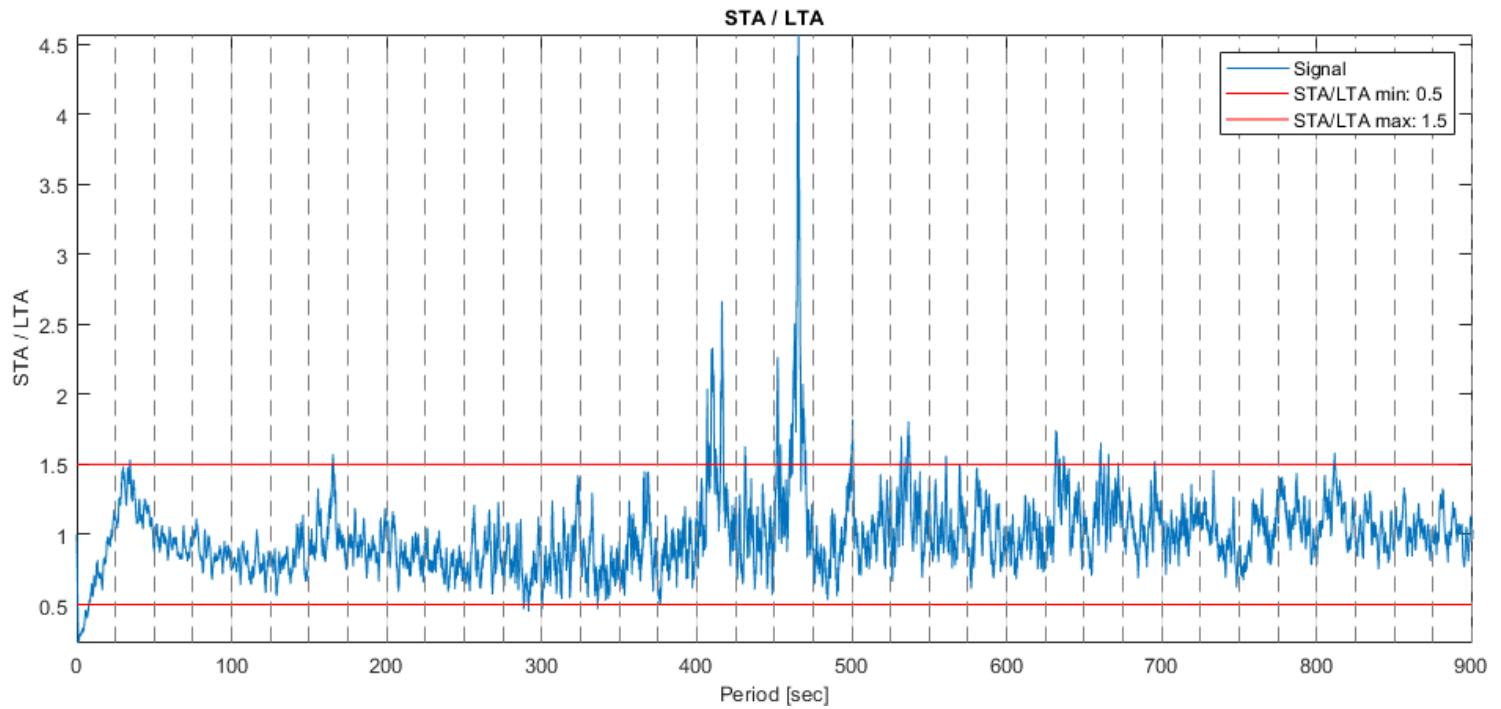
$$\text{radio}_i = \frac{STA}{LTA}$$

Se trabajará con el vector de registro normalizado, donde el promedio es cero y la desviación estándar es la unidad.

$$\left[\frac{F_0}{R_C} \right]_V \left[\frac{F_0}{R_C} \right]_p = \left[N \left[\frac{F_0}{R_C} \right]_k = 1 \left[\frac{F_0}{R_F} \right]_V k \left[\frac{F_0}{R_F} \right]_p \right] 1/p$$

```
V=zscore(Vo);
V=V/max(abs(V));
T=(0 : 1/Fs : length(V)/Fs-1/Fs)';
[ STA_LTA , STA , LTA ] = radio_STA_LTA( Fs , V , STA , LTA , T , vent ,
vmin , vmax );
```

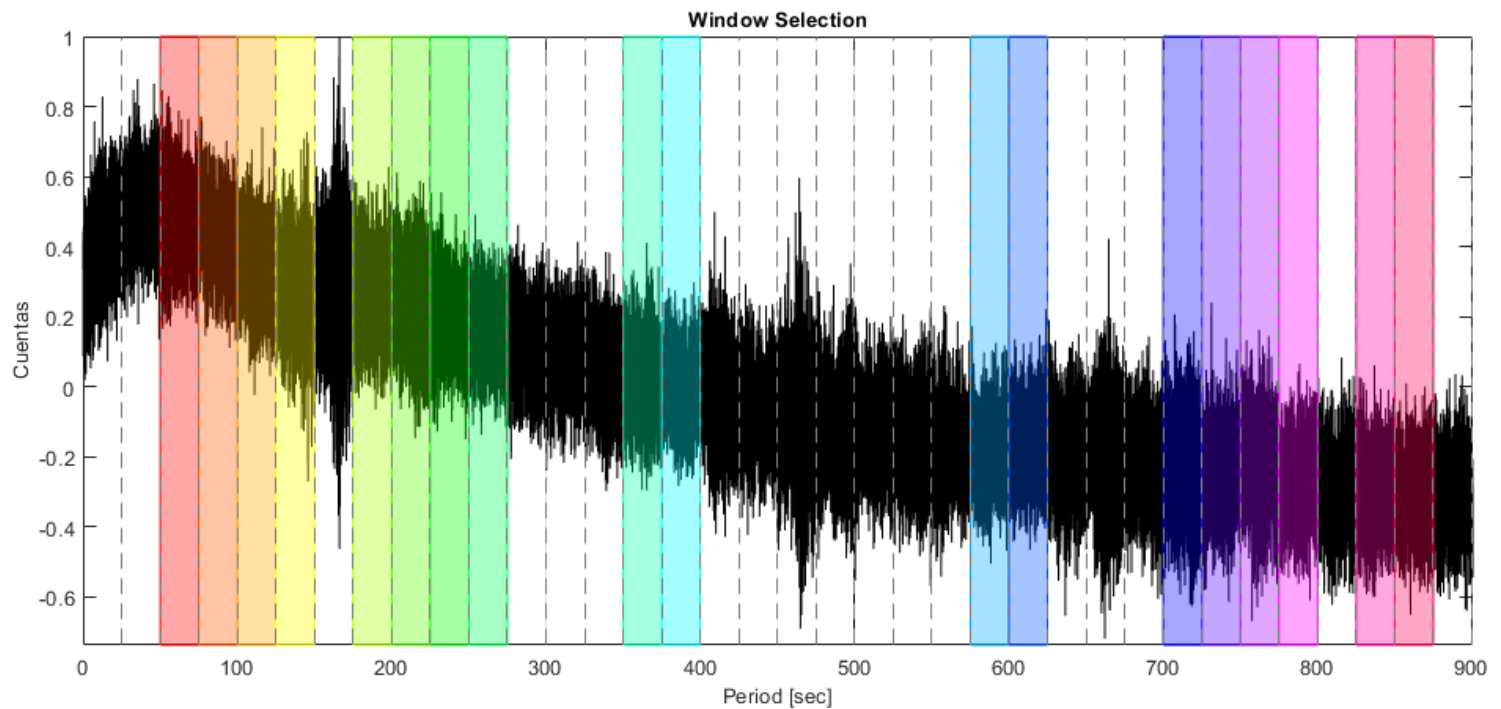




4. Algoritmo de seleccion de ventanas

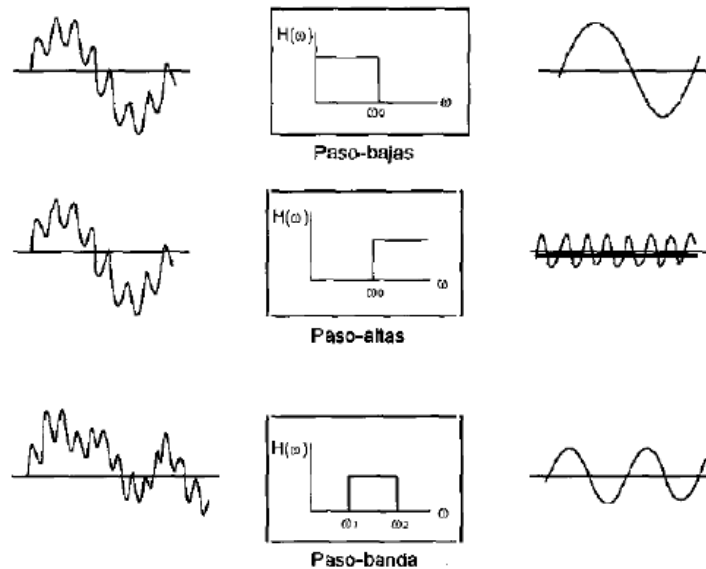
Con base a lo expresado en el algoritmo STA/LTA se realiza la selección de las ventanas que servirán para la extracción de vectores de datos y la caracterización de la estructura analizada.

```
[ MT , MV] = ventanas( Fs , T , V , STA_LTA , vent , vmin , vmax);
```



5. Algoritmo de aplicación de filtro

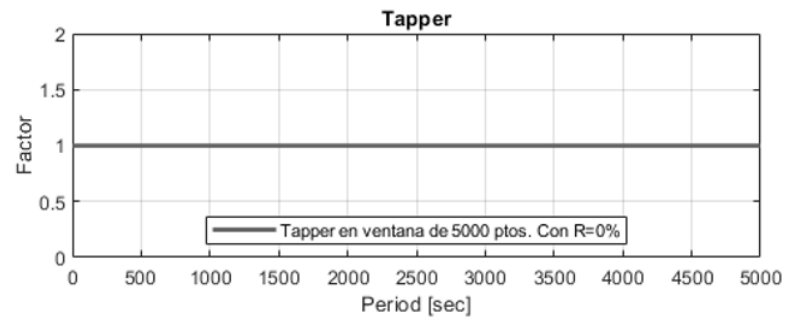
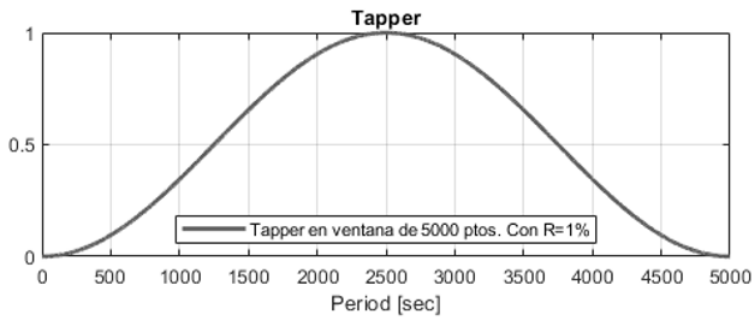
Se aplica un filtrado de respuesta finita (Finite Impulsive Response) de orden 128 con banda de paso en entre dos frecuencias que caractericen la estructura analizada. Se eliminan en las señales los picos que relacionan ruido no deseado relacionado con eventos ajenos al comportamiento de la estructura. Habrá que tener en cuenta la repercusión que el ruido tiene sobre la señal estudiada, eEste ruido puede estar causado tanto por fenómenos naturales como por el tratamiento de los datos. Así, podremos considerar que el acelerograma es la suma de la señal sísmica más el ruido. Puesto que no queremos que estos errores nos afecten, limitaremos la información a la banda de frecuencias deseada, en lugar de mantener información adicional a mayores y menores frecuencias, que pueden contener gran número de errores.



- **Filtro ideal paso-bajas:** elimina todas las frecuencias superiores a una dada, y deja pasar sin atenuación las frecuencias inferiores
- **Filtro ideal paso-altas:** elimina todas las frecuencias inferiores a una dada, y deja pasar sin atenuación las frecuencias superiores.
- **Filtro ideal paso-banda:** elimina todas las frecuencias inferiores a una dada, y las superiores a otra, y deja pasar sin atenuación las frecuencias comprendidas entre ellas.

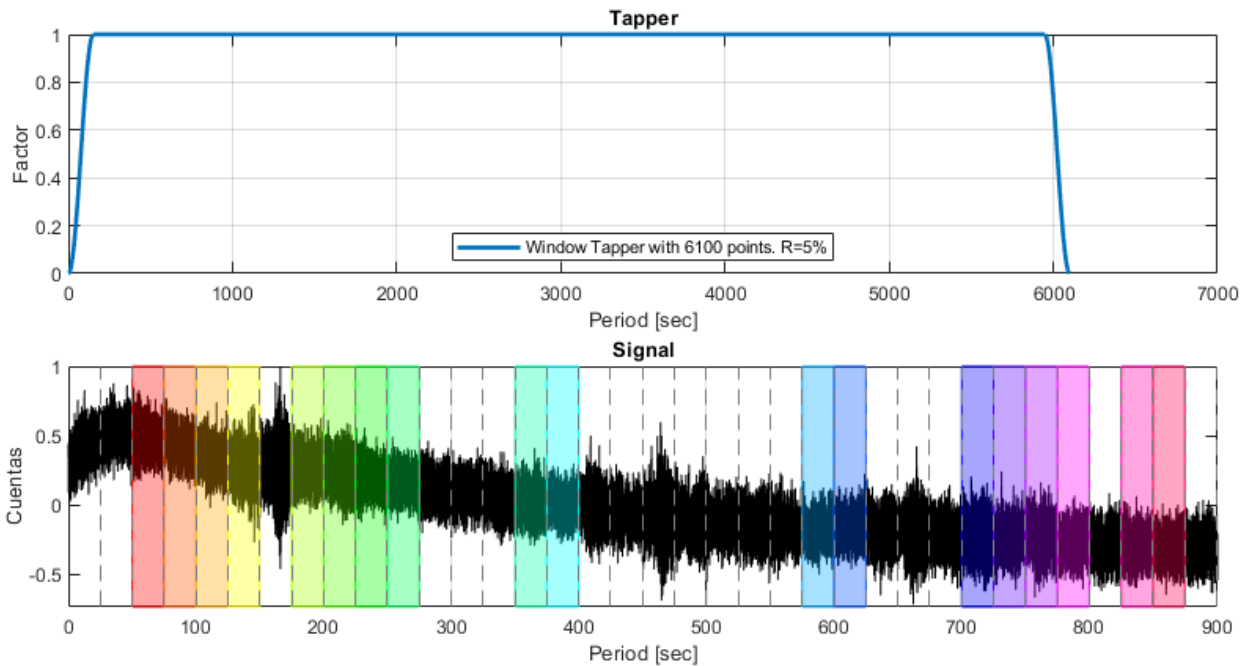
Ya que en la práctica los filtros ideales no se pueden utilizar ya que su función de transferencia en el tiempo es de longitud infinita, se utilizan aproximaciones a ellos. Entre estas aproximaciones se encuentran filtros tales como el Box car, Bartlett, Hanning, Butterworth frecuencial, entre otros.

El **tapper** consiste en multiplicar una ventana de señales previamente seleccionada por una función suave y nula en los extremos. Esto provoca que se evite el leakage que consiste en presentar altas frecuencias al inicio y final de las señales.



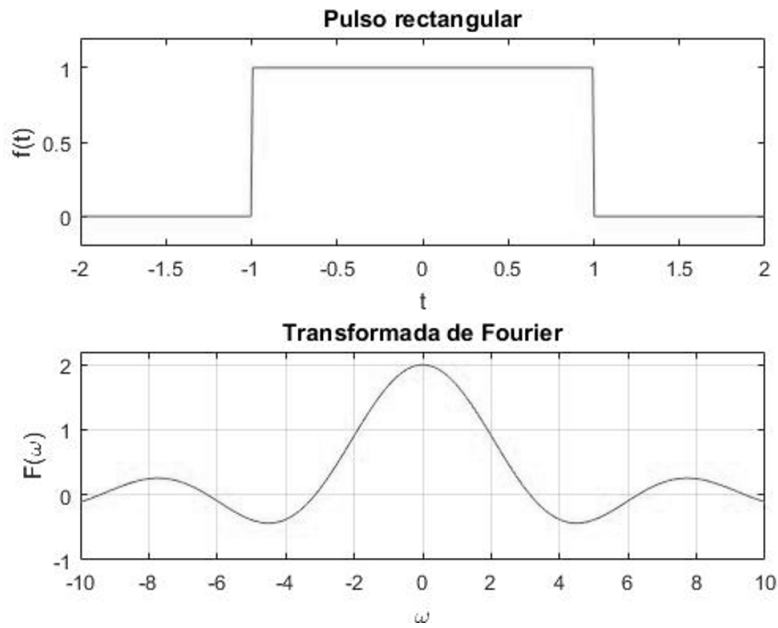
```
% Algoritmo TAPPER
[ MA , u ] = taper( MV , MT , p , vent , T , V );
```

ans = 18



6. Algoritmo para Transformada Rápida de Fourier

Se realiza el análisis donde se determinan las frecuencias predominantes del evento mediante un análisis utilizando el espectro de Fourier, estas frecuencias son aquellas para las cuales el espectro de Fourier de la aceleración alcanza su valor máximo. Este espectro de Fourier es un parámetro que proporciona un amplio conocimiento acerca del contenido en frecuencias de un acelerograma, de modo que para cada periodo de oscilación se puede presentar la amplitud máxima que le corresponde.



```
[ MFX , MFY , MFZ ]=fft_vent( Fs , MA , 'Y' , f1 , f2);
```

7. Algoritmo para suavizar (Konno & Ohmachi 1998)

La función de Konno y Ohmachi (1998) se utiliza para suavizar las funciones con el objetivo de aclarar el aspecto de las curvas, este método se basa en la función que se muestra en la siguiente ecuación trigonométrico-logarítmico.

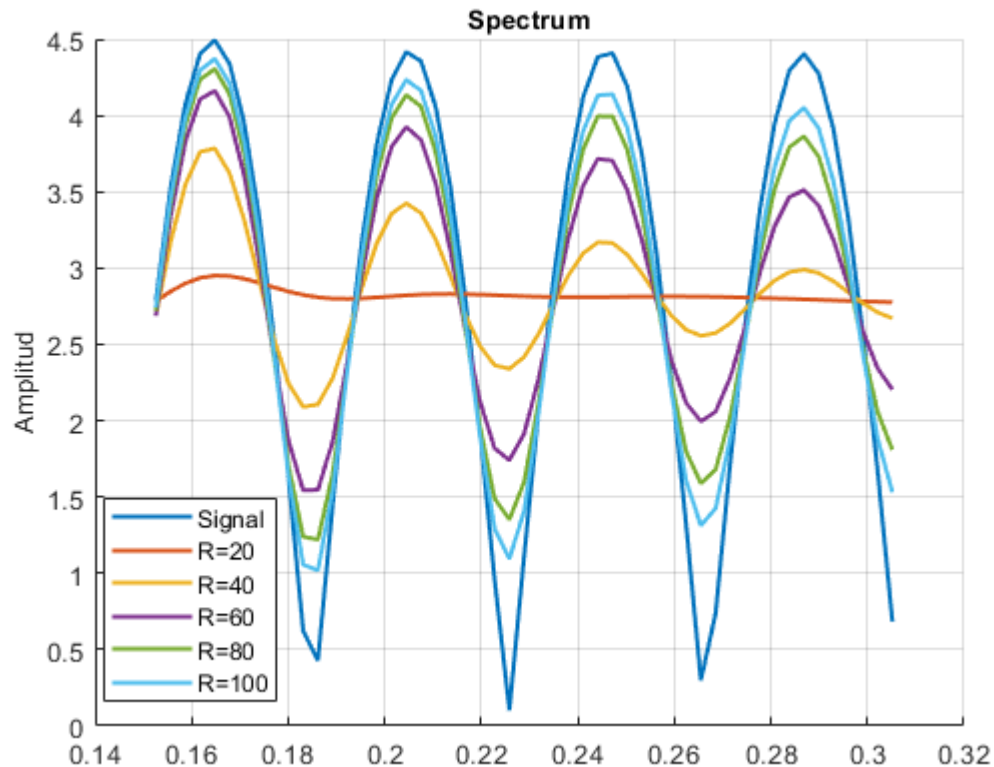
$$WB(f, f_i) = \left(\frac{\sin \left(b \log_{10} \left(\frac{f}{f_c} \right) \right)}{b \log_{10} \left(\frac{f}{f_c} \right)} \right)^4$$

Donde:

f: frecuencia

f_c : frecuencia central

b: coeficiente de ancho de banda



```
[ MFS ]=sua_vent( MFX , MFZ , bexp );
```

8. Algoritmo para Promedio

```
[ MFP ] = prom_vent(MFS);
```

9. Gráfica de Ventanas

9.1 Periodo predominante en ventanas

```
figure('color',[1 1 1],'numbertitle','off','name','Spectrums');
% Todos los espectros
subplot(2,1,1)
semilogx(MFX,MFS);
grid on
xlabel('Frequency [Hz]','fontsize',10); ylabel('Amplitude','fontsize',10);
title('Spectrum','fontsize',10);
% Espectro promedio
subplot(2,1,2)
semilogx(MFX(:,1),MFP,'linewidth',2);
grid on
xlabel('Frequency [Hz]','fontsize',10); ylabel('Amplitude','fontsize',10);
title('Average Spectrum','fontsize',10);
peaks0=findpeaks(MFP);peaks0=sort(peaks0,'descend');
peaks1=peaks0(1:4);
```

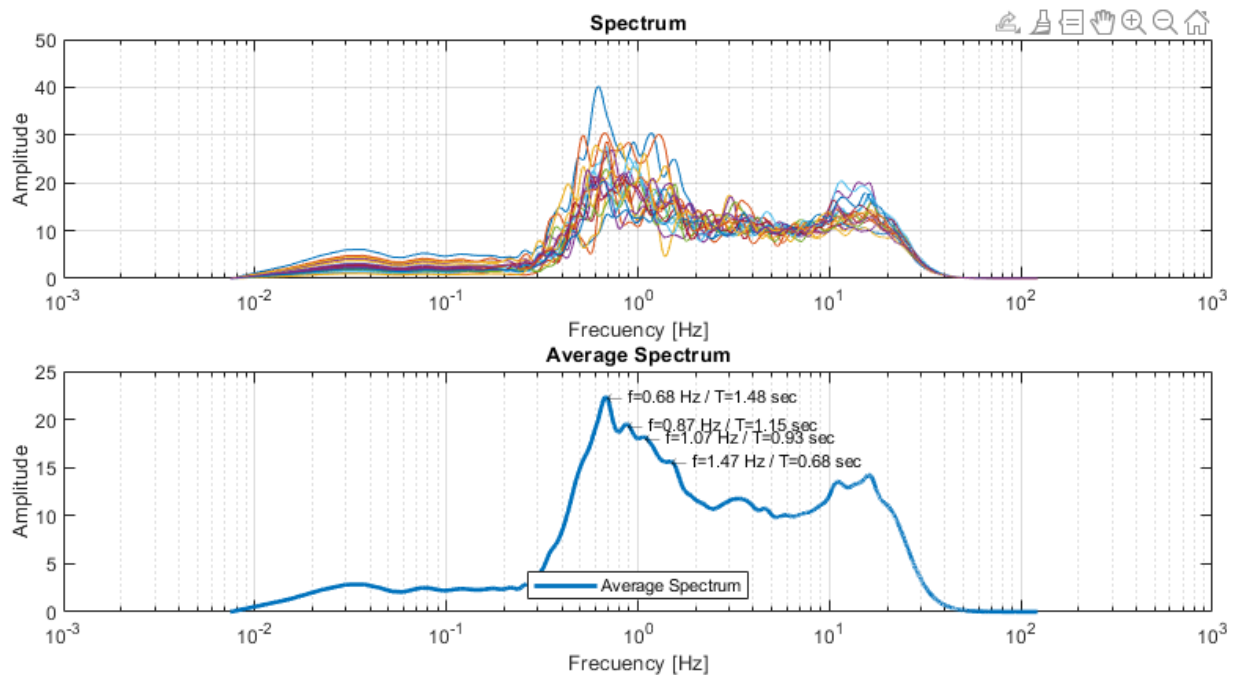
```

for i=1:1:size(peaks1,1)
    [p1(i,1),p2(i,2)]=find(peaks1(i)==MFP);
    % peaks1(i,2)=
    text97=['\leftarrow f=',num2str( round( MFX(p1(i,1)) , 2 ) ) , ' Hz /
T=',num2str( round( MFX(p1(i,1))^-1 , 2 ) ) , ' sec'];
    text( MFX(p1(i,1)) , MFP(p1(i,1)), text97,'fontsize',8)
end

legend('Average Spectrum', 'location','south');

% Propiedades de graficas
box on; ax = gca; ax.XGrid = 'on'; ax.YGrid = 'off';
fig=gcf; fig.Units='normalized';
fig.OuterPosition=[0.5 0.5 0.5 0.5];
set(gca, 'LooseInset', get(gca,'TightInset')); set(gcf,'Visible','on');

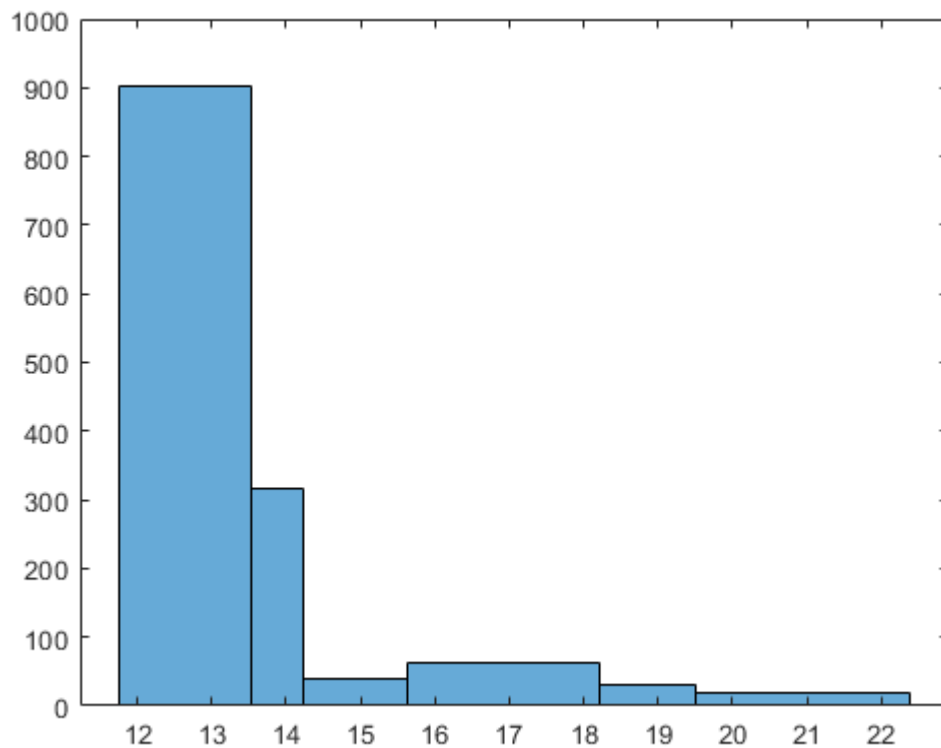
```



```

% Porcentaje de frecuencias
figure('color',[1 1 1],'numbertitle','off','name','Frequency content');
numfre=6;
peaks2=sort(peaks0,'ascend');
hist1=histogram(MFP,peaks2(end-numfre:end));

```



```

C1=100-(hist1.Values*100./sum(hist1.Values)); C1=sort(C1,'descend');
C2=(hist1.Values*100./sum(hist1.Values)); C2=sort(C2,'ascend');

C0=hist1.BinEdges;
COrden=sort(C0,'descend');

hb1=bar(C1);
hold on
hb1(1).FaceColor = [0.6 0.6 0.6];
hb1(1).FaceAlpha = 0.2;
for k = 1:1:numfre
    text(hb1.XData(k),hb1.YData(k),sprintf('%0.1f%%',hb1.YData(k)),...
        'HorizontalAlignment','center',...
        'VerticalAlignment','bottom');
    [p11(k,1),p22(k,2)]=find(COrden(k)==MFP);
    text96=['f=',num2str( round( MFX(p11(k,1)) , 2 ) ) , ' Hz /
T=',num2str( round( MFX(p11(k,1))^-1 , 2 ) ) , ' sec'];
    text( hb1.XData(k) , hb1.YData(k)/3,
text96,'fontsize',8,'Rotation',90,'color',[0 0 1])
end

hb2=bar(C2);
hb2(1).FaceColor = [0.1 0.1 0.1];
hb2(1).FaceAlpha = .5;

ylabel('Percentage [%]','fontsize',10);

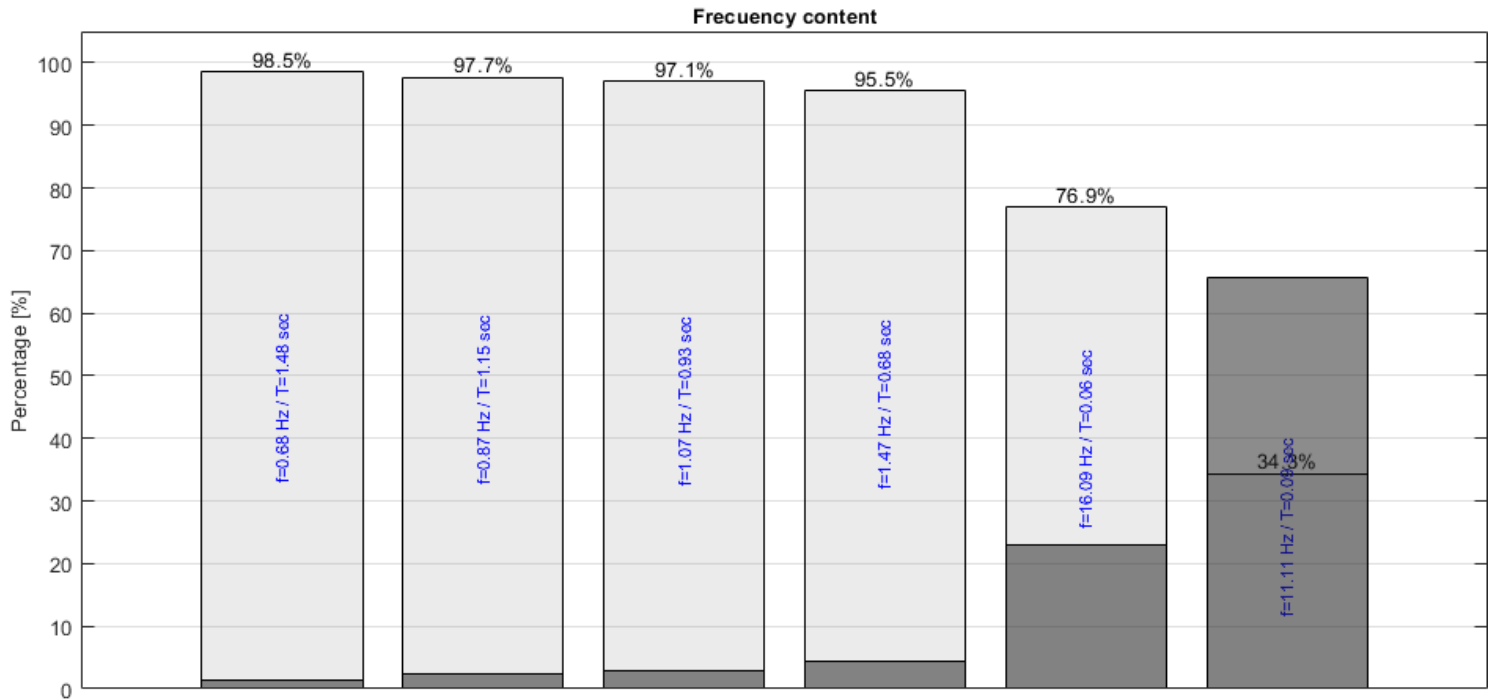
```

```

title('Frequency content','fontsize',10);
axis([0 numfre+1 0 105])
set(gca,'XTick',[]);

% Propiedades de graficas
box on; ax = gca; ax.XGrid = 'off'; ax.YGrid = 'on';
fig=gcf; fig.Units='normalized';
fig.OuterPosition=[0.5 0.5 0.5 0.5];
set(gca, 'LooseInset', get(gca,'TightInset')); set(gcf,'Visible','on');

```

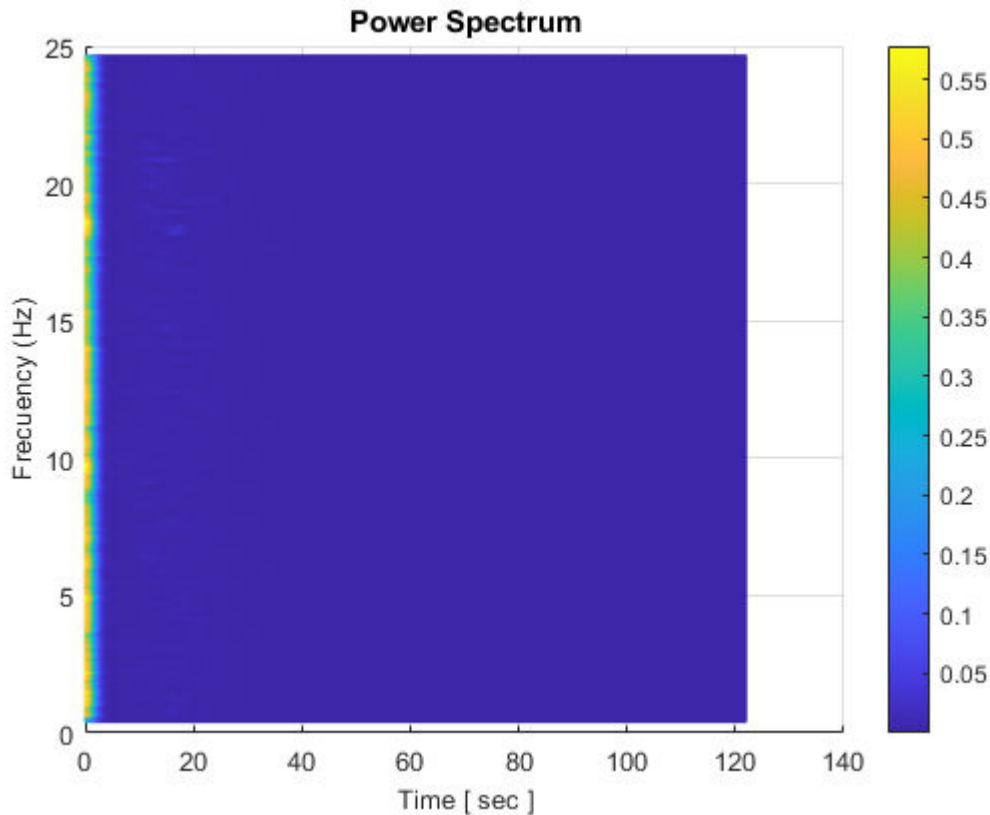


9.2 Espectro de potencia

```

[m,n]=size(MA);
figure('Color',[1 1 1],'NumberTitle','off','Name','Power Spectrum');
for i=1:1:size(MA,2)
    [p_sp,f_sp,t_sp] = pspectrum( MA(:,i) , Fs , 'spectrogram');
    hold on
    mesh(f_sp,t_sp,p_sp');
end
xlabel('Time [ sec ]','FontSize',10);
ylabel('Frequency (Hz)','FontSize',10);
zlabel('Power Spectrum','FontSize',10 );
title('Power Spectrum','FontSize',11);
grid on; hold off;
colorbar;

```



9.3 Escalograma de magnitud

Con la Transformada de Fourier con Ventana se logra una mejor localización de la aparición de una singularidad en una señal. Pero solo se conocerá en qué intervalo de tiempo se produce la singularidad, debido a que la localización depende del ancho elegido para la función ventana. Además, los eventos no podrán ser resueltos si aparecen muy cerca unos de otros, ya que no será posible distinguir diferentes comportamientos dentro de una misma amplitud de ventana

Una herramienta matemática que permite resolver estos problemas es la Transformada Wavelet. Este tipo de transformada es capaz de concentrarse en fenómenos transitorios y de alta frecuencia mejor que la Transformada de Fourier con Ventana. Con esta última, una vez que el tamaño de la ventana es elegido, todas las frecuencias son analizadas con las mismas resoluciones de tiempo y frecuencia, distinto de lo que sucede en la Transformada Wavelet que tiene un tamaño de ventana adaptado a las frecuencias.

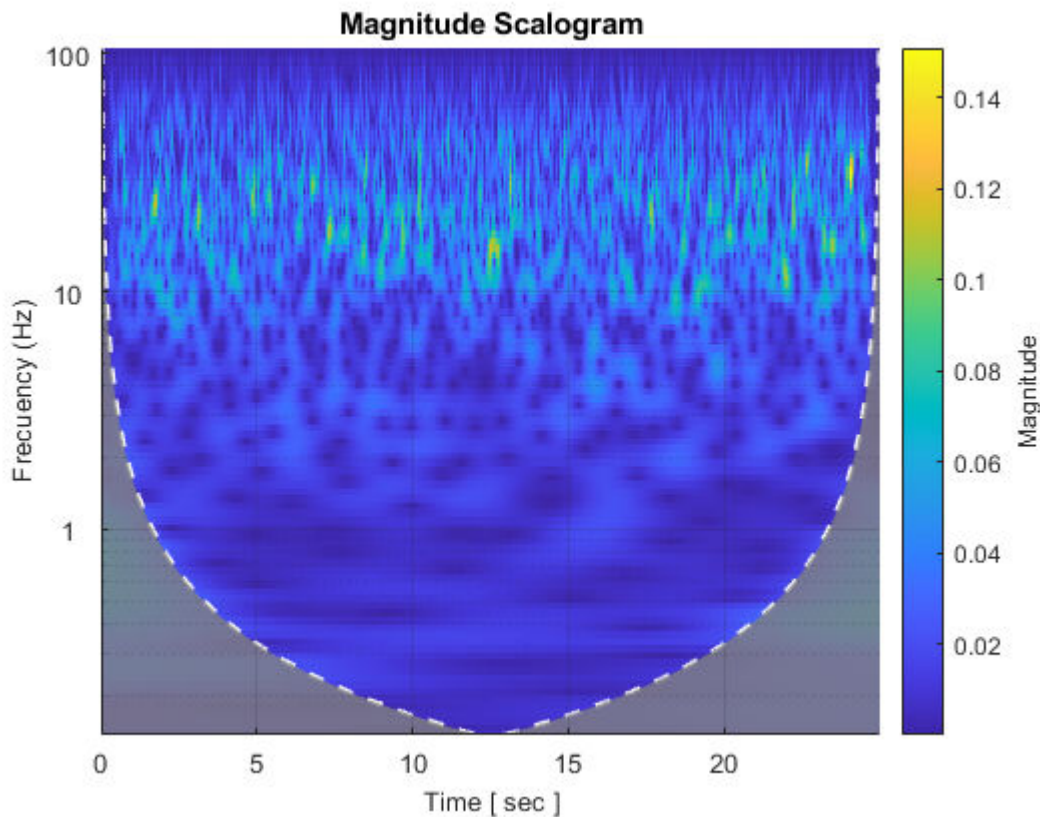
Esta Transformada es eficiente para el análisis local de señales no estacionarias y de rápida transitoriedad y, al igual que la Transformada de Fourier con Ventana, mapea la señal en una representación de tiempo-escala. El aspecto temporal de las señales es preservado. La diferencia está en que la Transformada Wavelet provee análisis de multiresolución con ventanas dilatadas. El análisis de las frecuencias de mayor rango se realiza usando ventanas angostas y el análisis de las frecuencias de menor rango se hace utilizando ventanas anchas

```
figure('Color',[1 1 1],'NumberTitle','off','Name','Magnitude Scalogram');
for i=1:1:size(MA,2)
    hold on
    cwt( MA(:,i) ,Fs)
```

```

end
xlabel('Time [ sec ]','FontSize',10);
ylabel('Frecuency (Hz)','FontSize',10);
title('Magnitude Scalogram','FontSize',11);
grid on; hold off;

```



10. Amplificación dinámica

9.1 Vibración Armónica Amortiguada.

```

%% Graph
figure('Color',[1 1 1],'NumberTitle','off','Name','Resonance');
% Índice de amortiguación
%%
zheta=0.05;DynamicAmplification( zheta );

```

Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize your function to return an output with the same size and shape as the input arguments.

```
zheta=0.10;DynamicAmplification( zheta );
```

Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize your function to return an output with the same size and shape as the input arguments.

```
zheta=0.20;DynamicAmplification( zheta );
```

Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize your function to return an output with the same size and shape as the input arguments.

```
zheta=0.50;DynamicAmplification( zheta );
```

Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize your function to return an output with the same size and shape as the input arguments.

```
zheta=0.70;DynamicAmplification( zheta );
```

Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize your function to return an output with the same size and shape as the input arguments.

```
zheta=1.00;DynamicAmplification( zheta );
```

Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize your function to return an output with the same size and shape as the input arguments.

```
DynamicAmplification( zheta )
```

Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize your function to return an output with the same size and shape as the input arguments.

```
legend( '\xi=5.00%', '\xi=10.0%', '\xi=20.0%', '\xi=50.0%', '\xi=70.0%', '\xi=100.0%', 'Location', 'East' );
```

```
% Propiedades de graficas
```

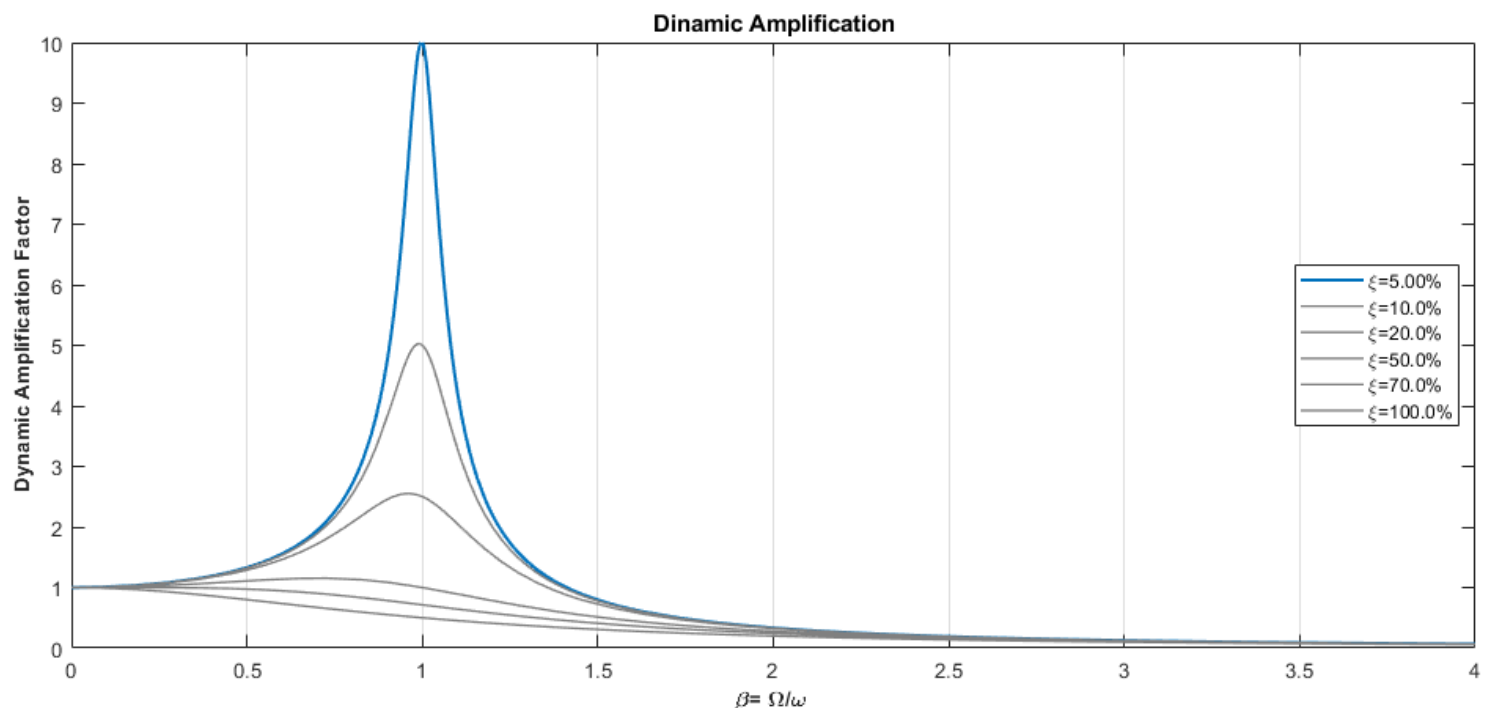
```
box on; ax = gca; ax.XGrid = 'on'; ax.YGrid = 'off';
```

```
fig=gcf; fig.Units='normalized';
```

```
fig.OuterPosition=[0.5 0.5 0.5 0.5];
```

```
set(gca, 'LooseInset', get(gca, 'TightInset')); set(gcf, 'Visible', 'on');
```

```
axis([0 4 0 10])
```



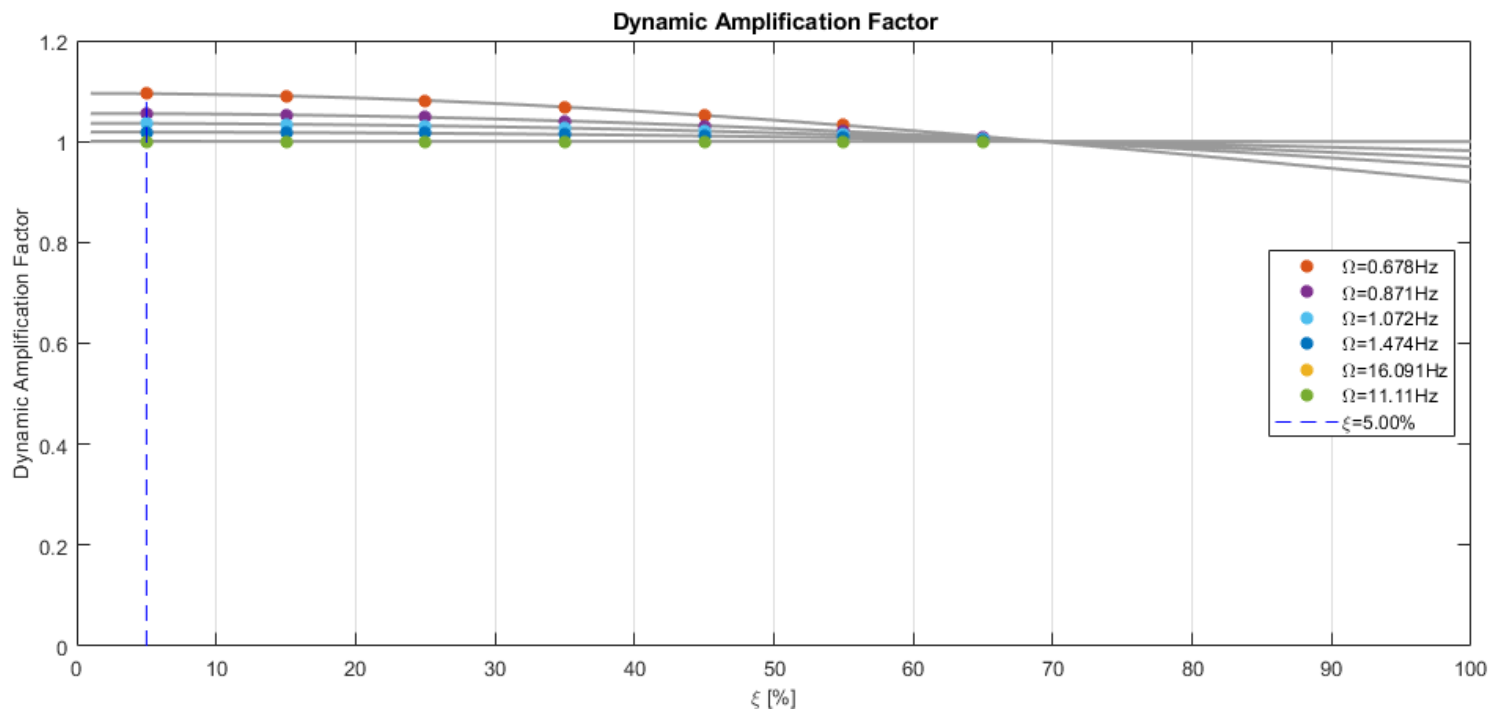
```

zheta=[ 0.05 , 0.15 , 0.25 , 0.35 , 0.45 , 0.55 , 0.65];
beta=(MFX(p11)./Testr).^-1;
aux1=MFX(p11);
zheta1=[ 0.01:0.01:1];
figure('Color',[1 1 1],'NumberTitle','off','Name','Dynamic Amplification
Factor');
hold on
for i=1:1:size(beta,1)
    Rd(:,i)=1./((1-beta(i,1)^2)^2 + (2*zheta1(1,:).*beta(i,1)).^2).^0.5;
    Rd_Data(:,i)=1./((1-beta(i,1)^2)^2 + (2*zheta(1,:).*beta(i,1)).^2).^0.5;
    plot1(i)=plot(zheta1*100',Rd(:,i),'color',[0.6 0.6 0.6],'linewidth',1.5);
    plot2(i)=plot(zheta*100',Rd_Data(:,i),'*','linewidth',2);
    lgn.txt{i}=['\Omega=',num2str(round(aux1(i),3)),'Hz'];
end

xlabel('\xi [%]','FontSize',10);
ylabel('Dynamic Amplification Factor','FontSize',10);
title('Dynamic Amplification Factor','FontSize',11);
plot3=plot([0.05 0.05]*100,[0 max(max(Rd))], 'b--');
legend([plot2';plot3],[lgn.txt, '\xi=5.00%'],'Location','East');

% Propiedades de graficas
box on; ax = gca; ax.XGrid = 'on'; ax.YGrid = 'off';
fig=gcf; fig.Units='normalized';
fig.OuterPosition=[0.5 0.5 0.5 0.5];
set(gca, 'LooseInset', get(gca, 'TightInset')); set(gcf, 'Visible', 'on');

```



Referencias Bibliograficas

- Jimenez, C. (2007). Procesamiento Digital de Señales Sísmicas con Matlab. *Revista de Investigación de Física*, 10(02)
- Konno, K., & Ohmachi, T. (1998). Ground-motion characteristics estimated from spectral ratio between horizontal and vertical components of microtremor. *Bulletin of the Seismological Society of America*, 88(1), 228–241.
- Parra, K., & Sarango, J. (2016). *ANÁLISIS ESTADÍSTICO DEL PERIODO EXPERIMENTAL DE VIBRACIÓN EN EDIFICIOS APORTICADOS DE HORMIGÓN ARMADO EN EL DISTRITO METROPOLITANO DE QUITO*. ESCUELA POLITÉCNICA NACIONAL.
- Peña, P. (2014). Algoritmo de Detección de Ondas P Invariante de Escala: Caso de Réplicas del Sismo del 11 de Marzo de 2010. Doctoral dissertation, Universidad de Chile, Geología, Región Metropolitana de Santiago.
- Pérez, A. (2018). *DETECCIÓN AUTOMÁTICA DE EVENTOS SÍSMICOS EN EL VOLCÁN COTOPAXI MEDIANTE TÉCNICAS DE APRENDIZAJE DE MÁQUINAS* [UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE]. <https://repositorio.espe.edu.ec/bitstream/21000/13812/1/T-ESPE-057527.pdf>
- Pérez, L., Cuartas, J., & Ferrer, C. (2006). Desarrollo de un sistema de adquisición y análisis de señales sísmicas para el observatorio sismológico de la Universidad del Quindío. Evento: XI Simposio de Tratamiento Digital de Señales, Imágenes y Visión Artificial (STSIVA), Colombia.