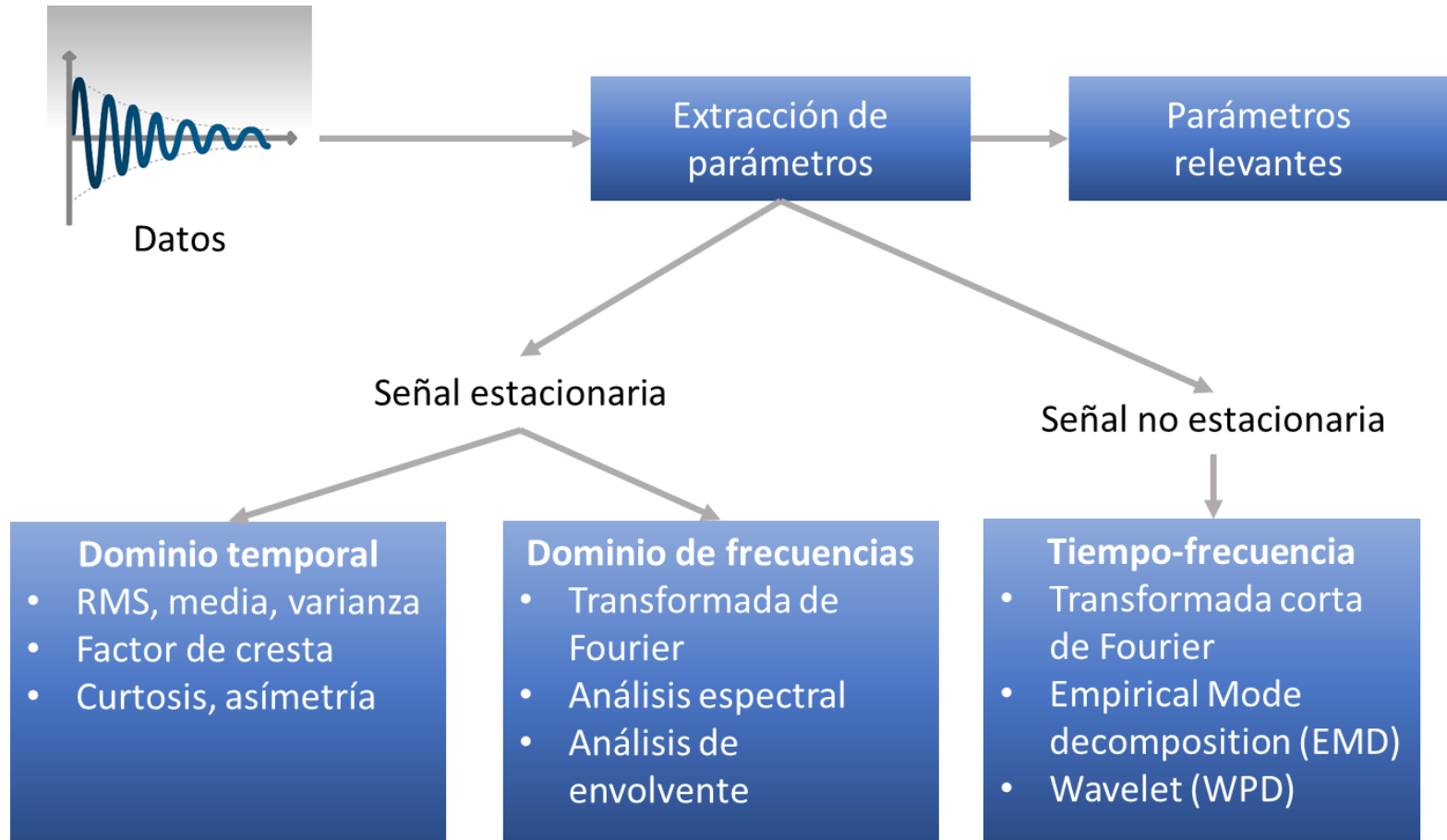


# Procesamiento de señales y aprendizaje de máquinas en mantenimiento predictivo

Viviana Meruane N.

Clase 4

# Extracción de parámetros



# Métodos tiempo-frecuencia

- Transformada corta de Fourier (espectrograma)
- Transforma de Wavelet (escalograma)

# Extracción de parámetros

## Transformada corta de Fourier

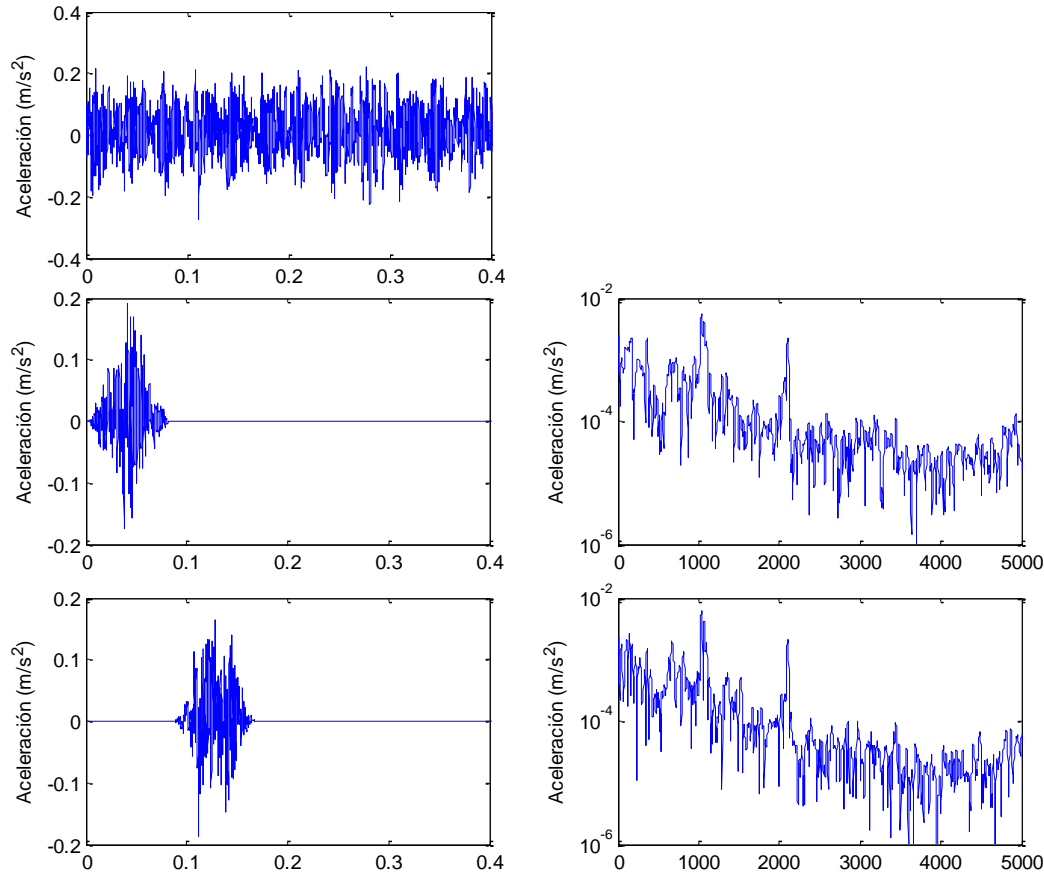
La transformada corta de Fourier se utiliza para señales no estacionarias y consiste en aplicar la transformada de Fourier a una ventana de datos “deslizante”.

El resultado es una representación en el tiempo del espectro en frecuencia de la señal.

Es importante la selección del ancho de la ventana, ya que define la resolución en el tiempo y en frecuencia.

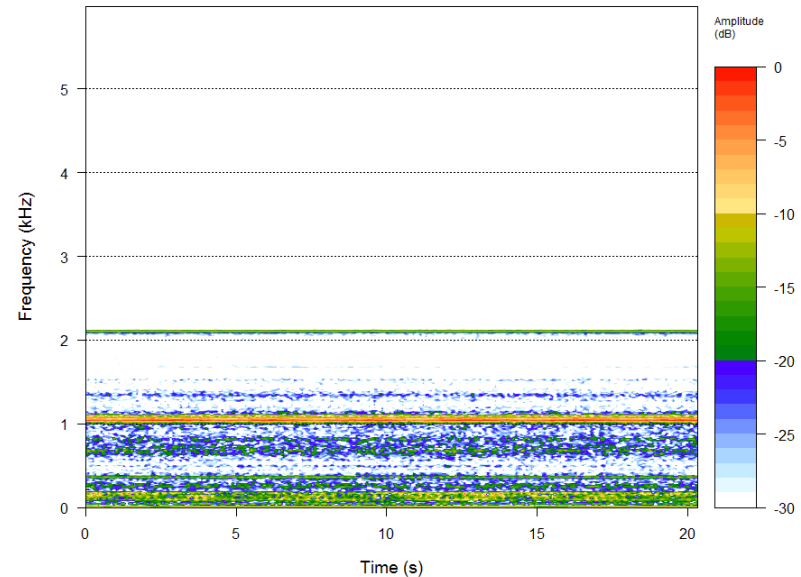
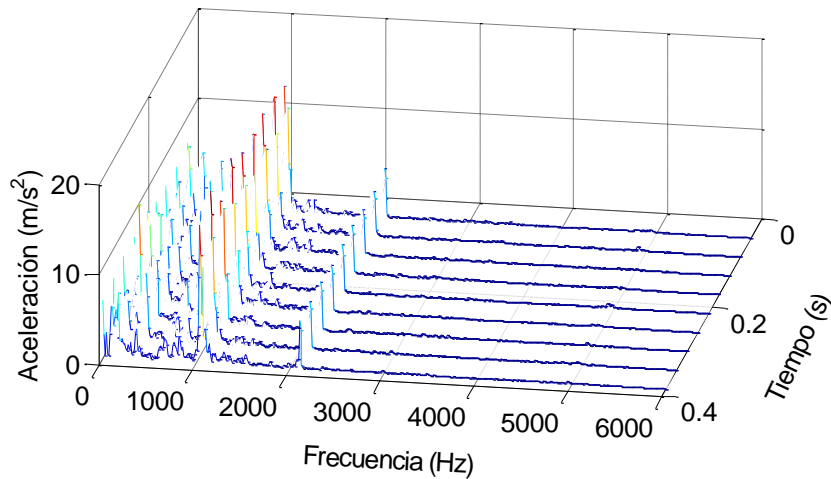
# Extracción de parámetros

## Transformada corta de Fourier



# Extracción de parámetros

## Transformada corta de Fourier



La segunda imagen se conoce como espectrograma

# Ejemplo 1

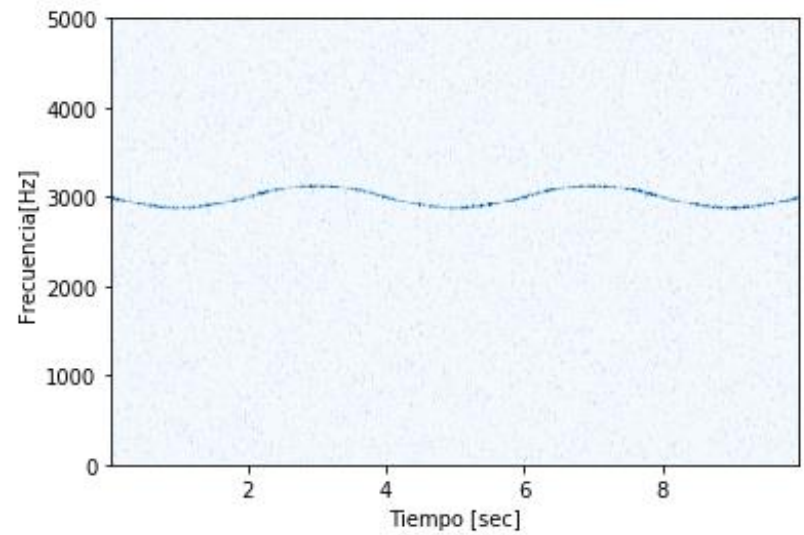
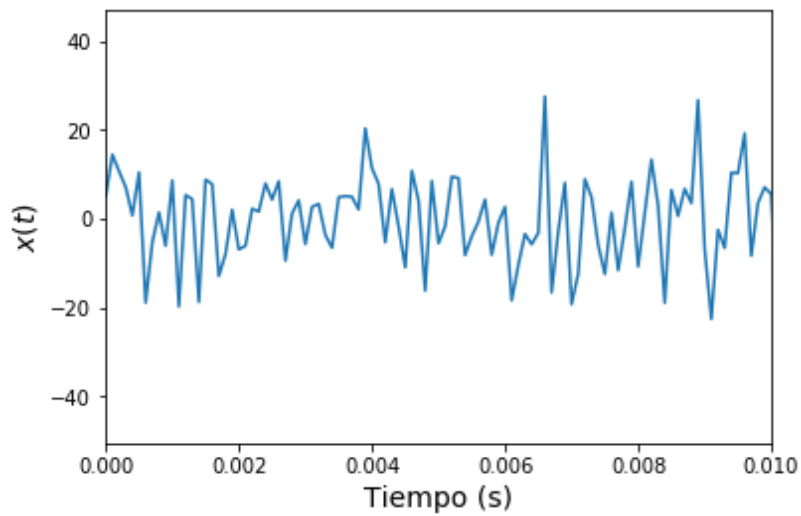
```
from scipy import signal
import matplotlib.pyplot as plt
import numpy as np
from numpy import pi
```

```
fs = 10e3 #frecuencia de muestreo Hz
N = 100000 #numero de datos
dt = 1/fs # Espaciado temporal
t = np.linspace(0, (N-1)*dt, N) # Intervalo de tiempo en segundos
mod = 500*np.cos(2*np.pi*0.25*t)
carrier = 5 * np.sin(2*np.pi*3e3*t + mod)
mu, sigma = 0, 10
noise = np.random.normal(mu,sigma,N)
x = carrier + noise
```

```
plt.plot(t, x)
plt.xlabel('Tiempo (s)', fontsize=14)
plt.ylabel('$x(t)$', fontsize=14)
plt.xlim(0,0.01)
plt.show()
```

```
f, t, Sxx = signal.spectrogram(x, fs,window='hann',nperseg=500,noverlap=250)
plt.figure()
plt.pcolormesh(t, f, Sxx,cmap='Blues')
plt.ylabel('Frecuencia[Hz]')
plt.xlabel('Tiempo [sec]')
plt.show()
```

# Ejemplo 1





# Ejemplo 2

```
#importar librerías
import scipy.io as sio
from scipy import signal
import matplotlib.pyplot as plt
import numpy as np
from numpy import pi
import math
from scipy.fftpack import fft, fftfreq
```

```
#Leer datos
Datos0=sio.loadmat('normal.mat')
Datos1=sio.loadmat('outer.mat')
Datos2=sio.loadmat('inner.mat')
Normal=Datos0['normal']
Outer=Datos1['outer']
Inner=Datos2['inner']
```

```
#Datos
Fs=48828 #sampling rate
Nd=20000
```

# Ejemplo 2

```
x1=Normal[1:Nd,0]
f, t, S1 = signal.spectrogram(x1,Fs>window='hann',nperseg=500,noverlap=250)
```

```
x2=Outer[1:Nd,0]
f, t, S2 = signal.spectrogram(x2,Fs>window='hann',nperseg=500,noverlap=250)
```

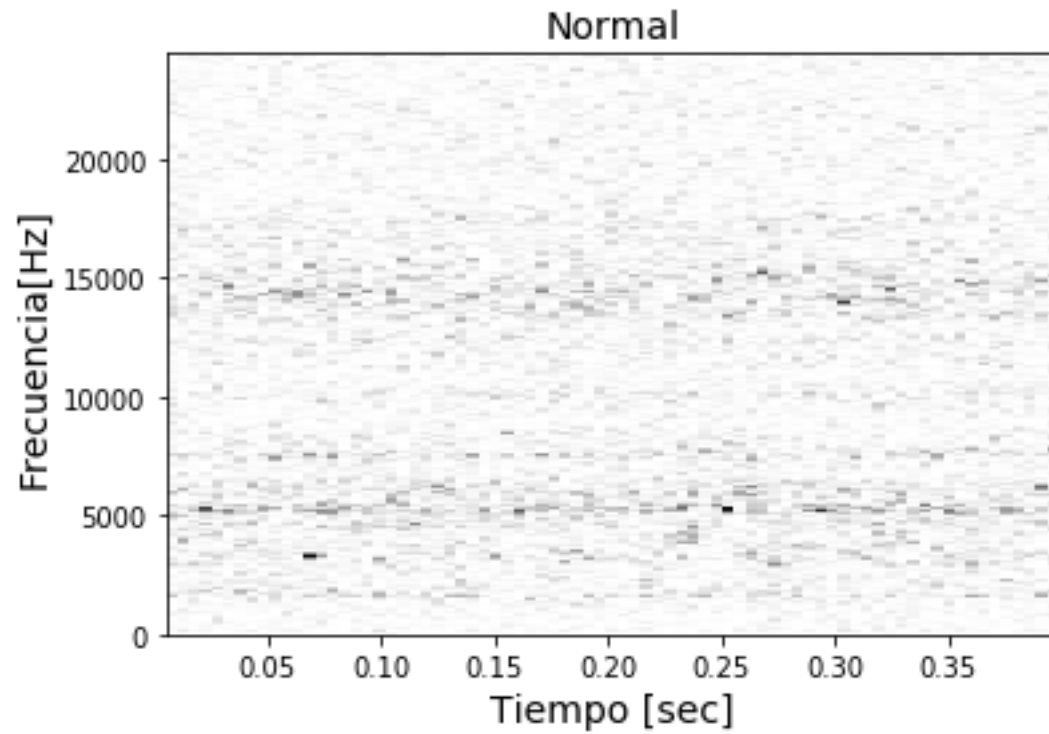
```
x3=Inner[1:Nd,0]
f, t, S3 = signal.spectrogram(x3,Fs>window='hann',nperseg=500,noverlap=250)
```

```
plt.pcolormesh(t, f, S1,cmap='binary')
plt.ylabel('Frecuencia[Hz]', fontsize=14)
plt.xlabel('Tiempo [sec]', fontsize=14)
plt.title('Normal', fontsize=14)
plt.show()
```

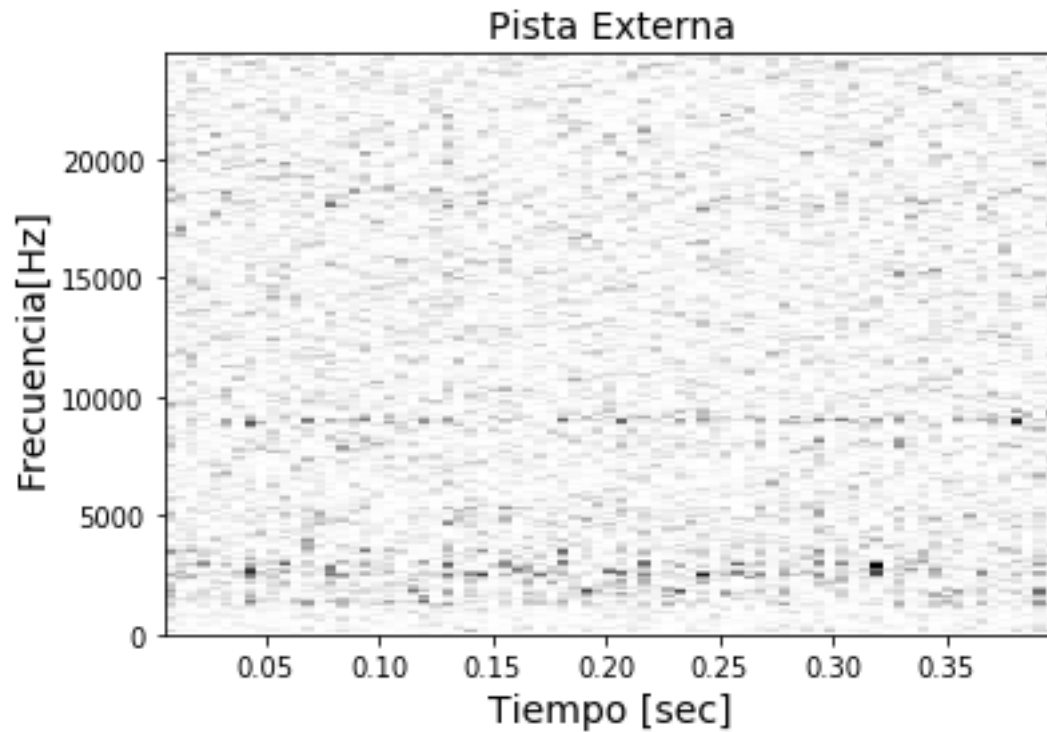
```
plt.pcolormesh(t, f, S2,cmap='binary')
plt.ylabel('Frecuencia[Hz]', fontsize=14)
plt.xlabel('Tiempo [sec]', fontsize=14)
plt.title('Pista Externa', fontsize=14)
plt.show()
```

```
plt.pcolormesh(t, f, S3,cmap='binary')
plt.ylabel('Frecuencia[Hz]', fontsize=14)
plt.xlabel('Tiempo [sec]', fontsize=14)
plt.title('Pista Interna', fontsize=14)
plt.show()
```

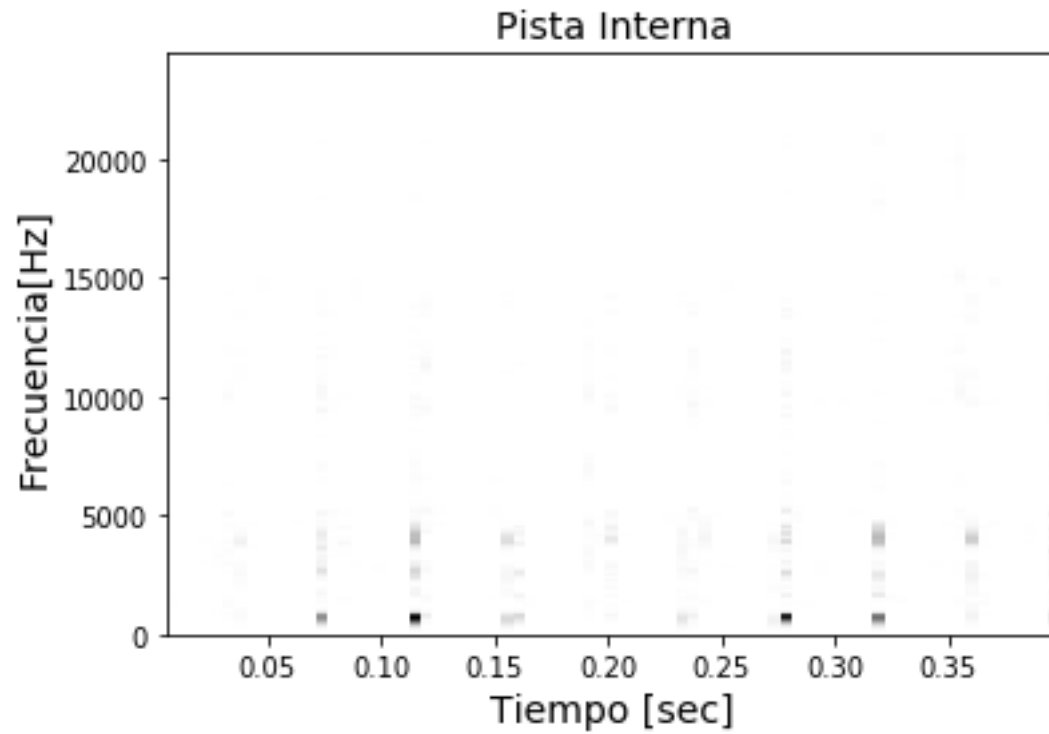
# Ejemplo 2



# Ejemplo 2



# Ejemplo 2



# Extracción de parámetros

## Wavelets

A diferencia de la transformada de Fourier donde se ocupan funciones base estacionarias (exponencial compleja), la transformada de Wavelet utiliza funciones base no estacionarias. Estas funciones se denominan “wavelets” y pueden dilatarse y desfasarse en función del tiempo.

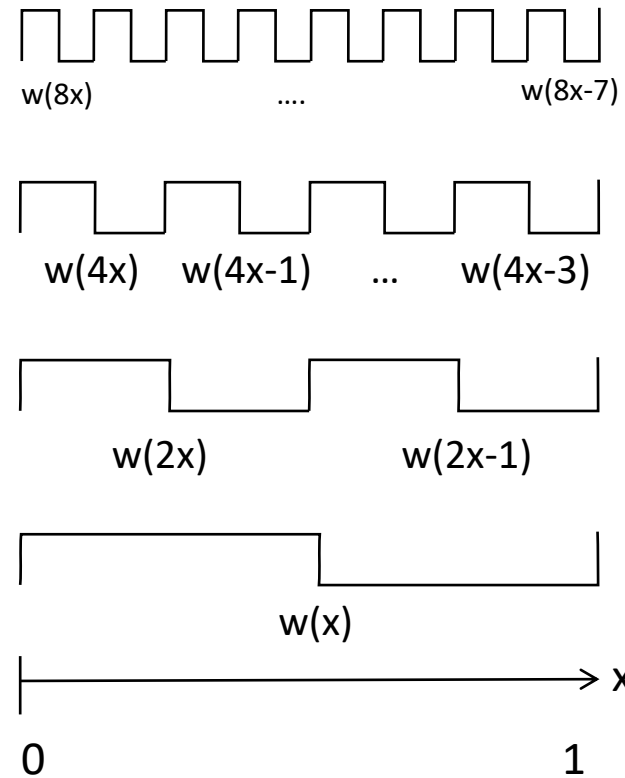
La principal ventaja es que la resolución en tiempo y frecuencia son independientes. Por lo tanto, se puede estudiar el contenido en frecuencia de la señal sin perder información en el dominio temporal.

# Extracción de parámetros

## Wavelets

- Familia de wavelets haar

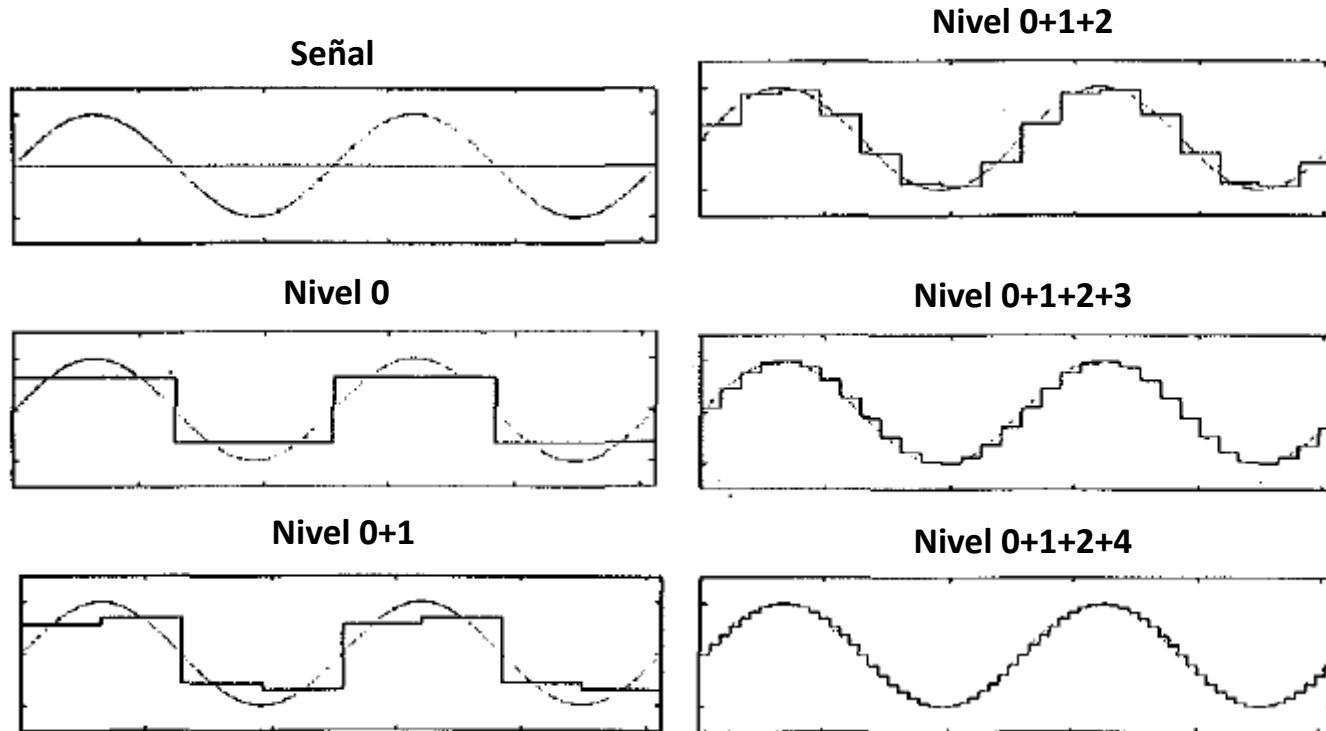
$$f(x) = a_0 + a_1 w(x) + a_2 w(2x) + a_3 w(2x - 1) + a_4 w(4x) + \dots a_9 w(8x - 7)$$



# Extracción de parámetros

## Wavelets

- Ejemplo con wavelets haar



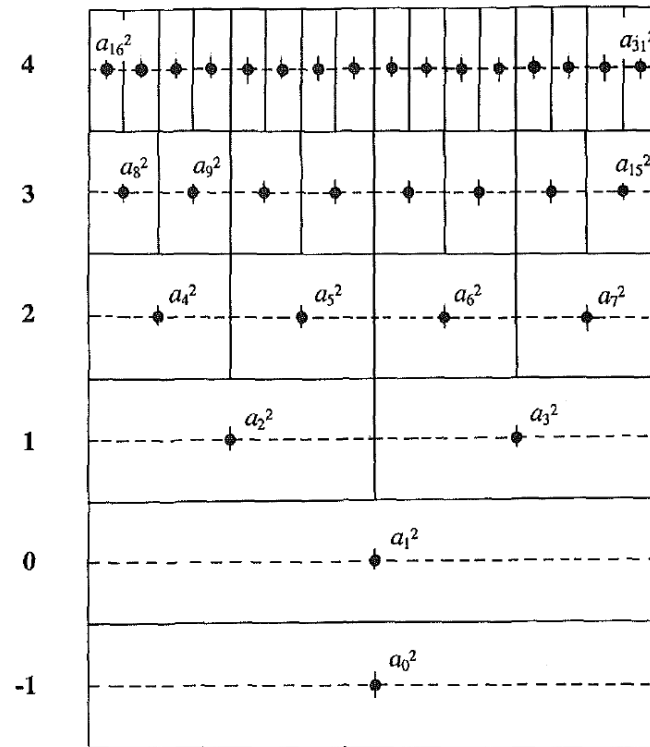


# Extracción de parámetros

## Wavelets

- Mapa de wavelets

Nivel



# Extracción de parámetros

## Wavelets

- Transformada continua de Wavelet

$$\psi_{u,s}(x) = \frac{1}{\sqrt{s}} \psi\left(\frac{x-u}{s}\right)$$

Donde  $s$  y  $u$  corresponden a parámetros de escala y traslación, respectivamente. La función  $\psi(x)$  es llamada madre wavelet.

# Extracción de parámetros

## Wavelets

Para una función  $f(x)$ , la transformada continua de wavelet (CWT) es el producto punto entre la señal y la familia de wavelets:

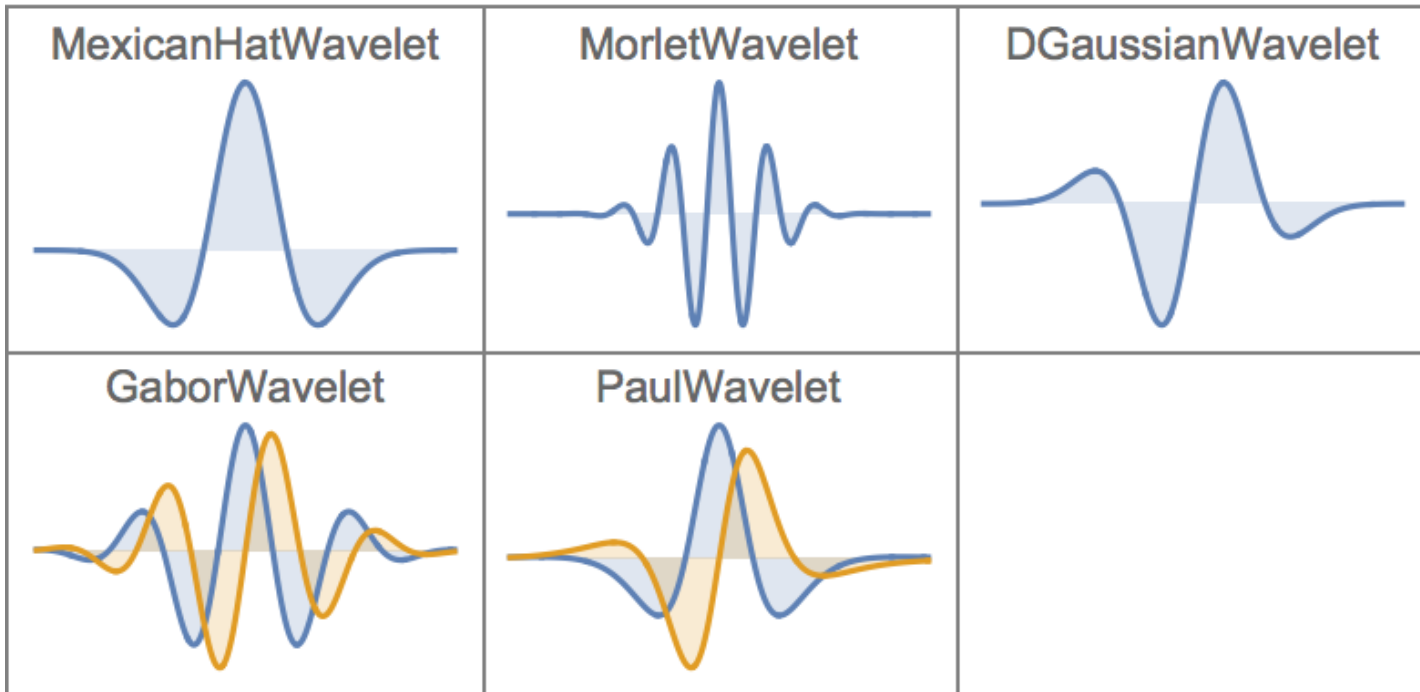
$$Wf(u, s) = \langle f, \psi_{u,s} \rangle = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(x) \psi\left(\frac{x-u}{s}\right) dx$$

Donde  $Wf(u, s)$  corresponde al coeficiente de wavelet.

# Extracción de parámetros

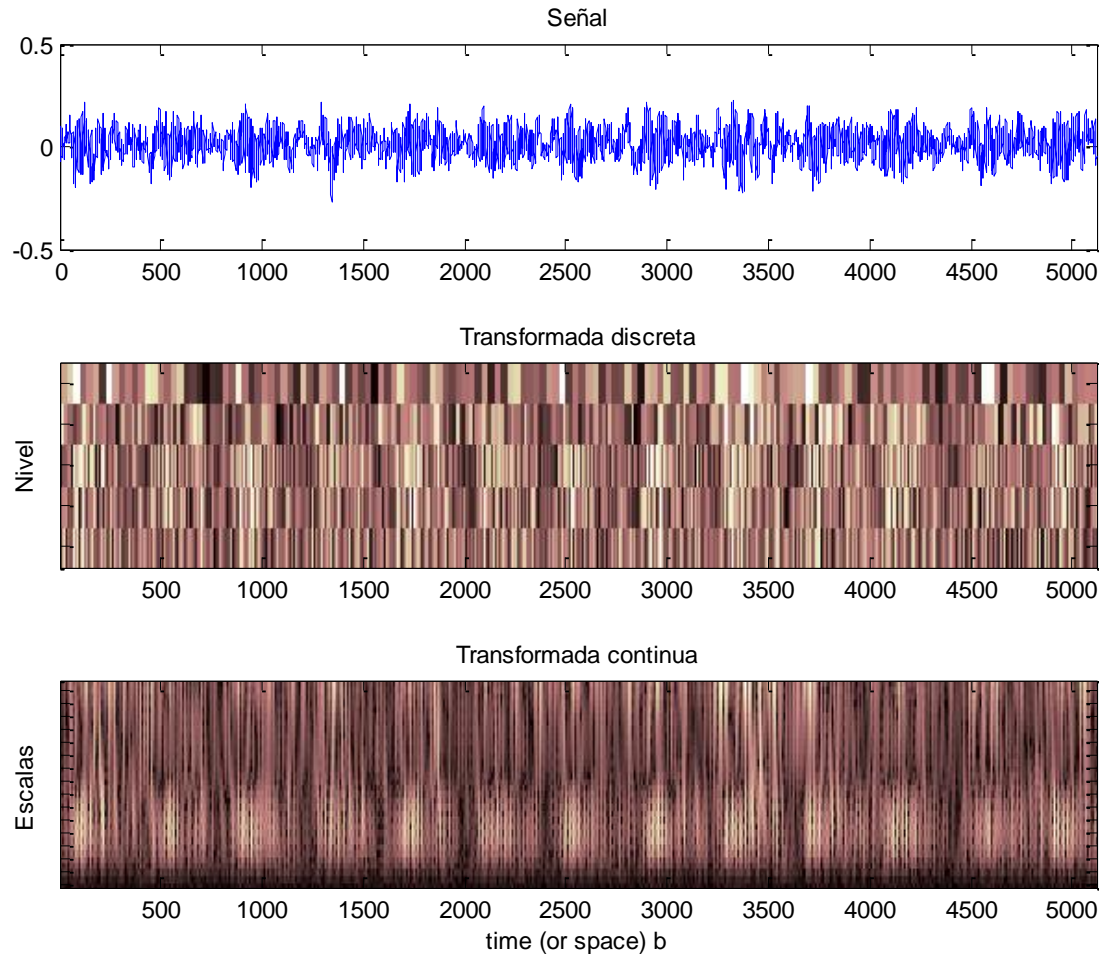
## Wavelets

- Tipos de wavelets



# Extracción de parámetros

## Wavelets



# Ejemplo 3

```
#importar librerias
import scipy.io as sio
import matplotlib.pyplot as plt
import numpy as np
import pywt
```

```
#Leer datos
Datos0=sio.loadmat('normal.mat')
Datos1=sio.loadmat('outer.mat')
Datos2=sio.loadmat('inner.mat')
Normal=Datos0['normal']
Outer=Datos1['outer']
Inner=Datos2['inner']
```

```
#Datos
Fs=48828 #sampling rate
dt=1/Fs
Nd=20000
```

```
scales=np.arange(1, 50)
```

```
x1=Normal[1:Nd,0]
x2=Outer[1:Nd,0]
x3=Inner[1:Nd,0]
t = np.linspace(0, (Nd-1)*dt, Nd) # Intervalo de tiempo en segundos
```

# Ejemplo 3

```
wavlist = pywt.wavelist(kind='continuous')
```

```
cwt1,frec1 = pywt.cwt(x1,scales,'gaus1')
```

```
cwt2,frec2 = pywt.cwt(x2,scales,'gaus1')
```

```
cwt3,frec3 = pywt.cwt(x3,scales,'gaus1')
```

```
plt.pcolormesh(t,scales,cwt1,cmap='binary')
```

```
plt.ylabel('Escala', fontsize=14)
```

```
plt.xlabel('Tiempo [sec]', fontsize=14)
```

```
plt.title('Normal', fontsize=14)
```

```
plt.show()
```

```
plt.pcolormesh(t,scales,cwt2,cmap='binary')
```

```
plt.ylabel('Escala', fontsize=14)
```

```
plt.xlabel('Tiempo [sec]', fontsize=14)
```

```
plt.title('Pista Externa', fontsize=14)
```

```
plt.show()
```

```
plt.pcolormesh(t,scales,cwt3,cmap='binary')
```

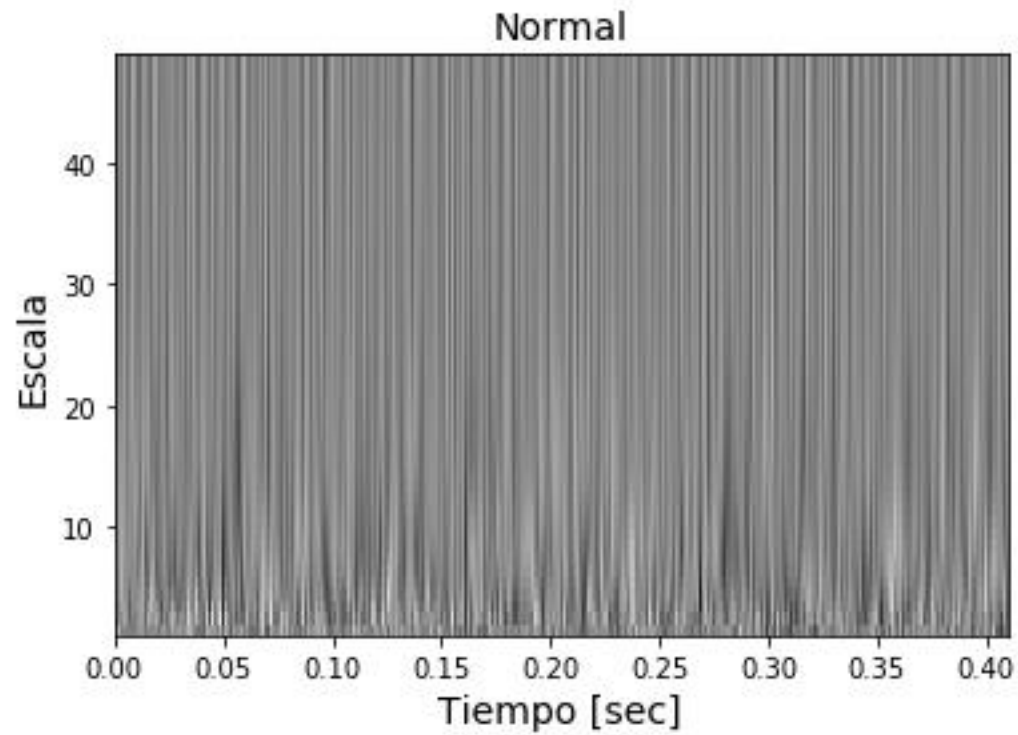
```
plt.ylabel('Escala', fontsize=14)
```

```
plt.xlabel('Tiempo [sec]', fontsize=14)
```

```
plt.title('Pista interna', fontsize=14)
```

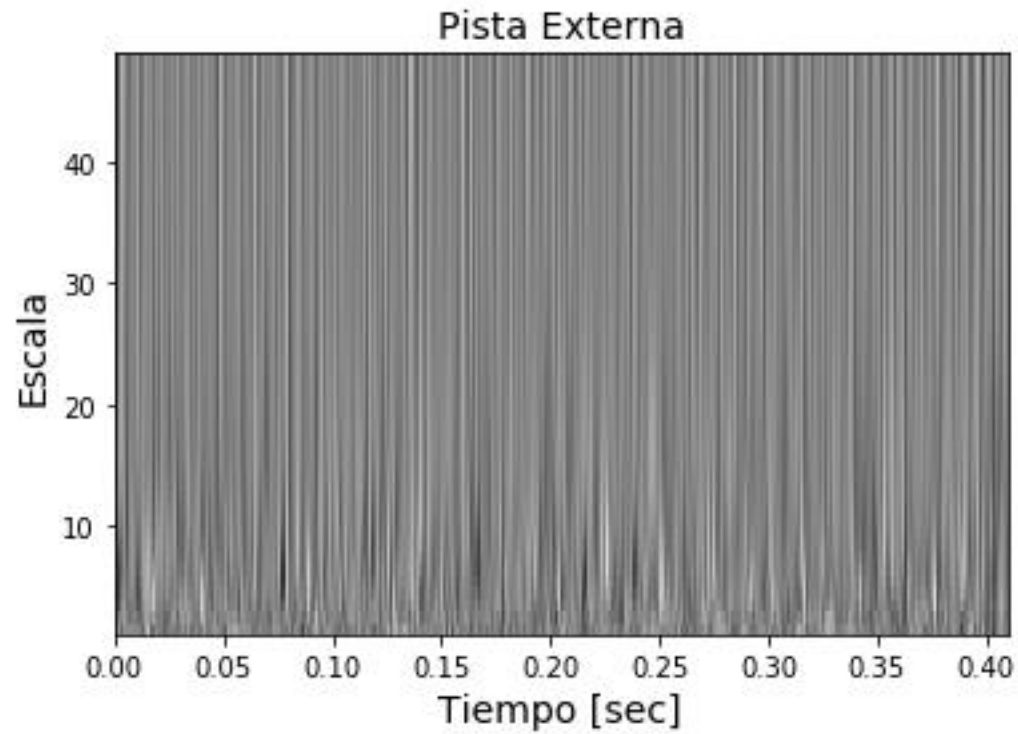
```
plt.show()
```

# Ejemplo 3

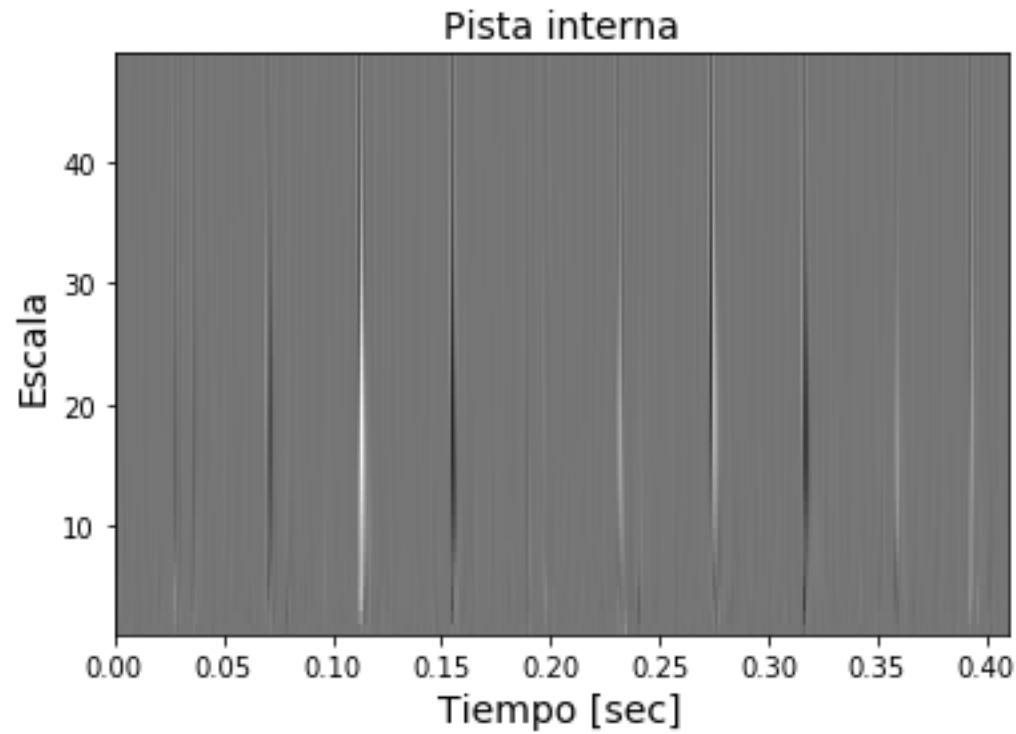




# Ejemplo 3



# Ejemplo 3



# Tarea

En la siguiente página se encuentra disponible un set de datos para un rodamiento sin falla, con falla en la pista externa, con falla en la pista interna, con falla en las bolas y con una combinación de fallas. Adicionalmente los datos consideran condiciones de operación variables:

<https://data.mendeley.com/datasets/v43hmbwxpm/2>

# Tarea

- Considere los cinco estados del rodamiento y la condición de operación: “increasing then decreasing speed”.
- Evalué las distintas opciones de extracción de parámetros vistas en clases (en el tiempo, frecuencia y tiempo-frecuencia), y concluya cuales son las más adecuadas para este caso.

# Tarea

- Nota: el canal 2 en los datos es la señal de un sensor encoder. Se puede deducir la velocidad de giro al calcular el tiempo de paso entre una ranura y otra, sabiendo que el encoder tiene 1024 ranuras por giro.