

# Procesamiento de señales y aprendizaje de máquinas en mantenimiento predictivo

Viviana Meruane N.

Clase 3

# Introducción

Hoy en día se hace cada vez más importante poder anticipar las fallas de un sistema de manera de planificar acciones de mantenimiento y así reducir costos y riesgos.

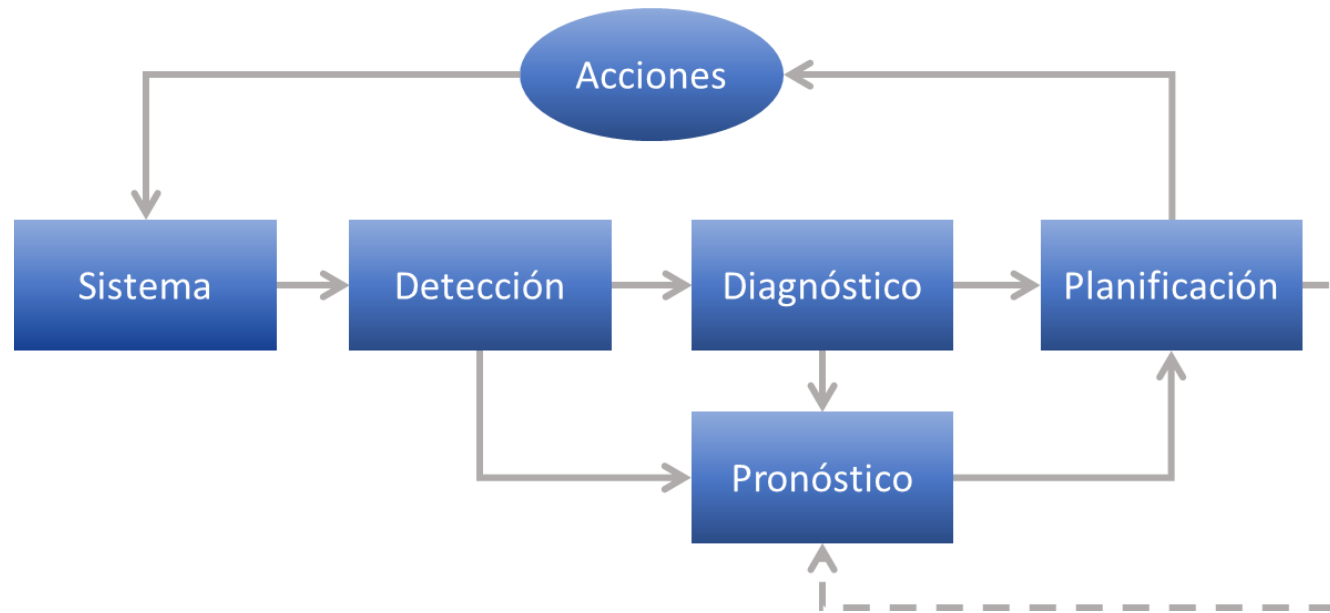
El desarrollo de técnicas de “Prognostics and Health Managment” (PHM) juega un rol cada vez más importante. Siendo el proceso de pronóstico uno de los temas principales de investigación a nivel mundial.

# Introducción

Una estrategia de PHM está compuesta de tres etapas principales:

1. Detección: Trata de identificar el modo de operación del sistema y su estado.
2. Diagnóstico: Cuando se detecta una anomalía, el diagnóstico identifica el componente que tiene la anomalía y su severidad (de los efectos a las causas).
3. Pronóstico: El pronóstico trata de predecir el estado futuro del sistema y su vida útil remanente (de las causas a los efectos).

# Introducción



Detección, diagnóstico, pronóstico y planificación en un sistema de mantenimiento inteligente.

Para lograr la detección, diagnóstico y pronóstico de las fallas de un sistema se requiere de un sistema adecuado de adquisición y procesamiento de los datos.

# Adquisición de datos

Estrategia general para la obtención de datos para PHM:



Sistema

1. Selección de los componentes críticos
2. Variables físicas a monitorear y sensores a instalar
3. Adquisición y pre-procesamiento de los datos

Datos  
para PHM

# Componentes críticos

Un componente crítico se define como un componente cuya falla lleva a una detención del sistema completo y cuya frecuencia de fallas es alta.



Sistema



1. División en subsistemas
2. Descripción técnica  
(componentes,  
relaciones funcionales,  
etc.)
3. Análisis de fallas:
  - Pareto
  - Árboles de falla
  - FMECA
  - HAZOP
  - ...



Componentes  
críticos

# Variables físicas a monitorear

El monitoreo del estado de salud de un componente se realiza mediante el análisis de algunos de sus parámetros físicos u operacionales. La selección de estos parámetros es muy importante ya que una mala selección puede llevar a la no detección o a falsas alarmas.

Sin embargo, esta selección requiere de un conocimiento profundo entre la evolución de la degradación y las variables que se ven afectadas.

Por ejemplo, las vibraciones axiales en un rodamiento entregan información sobre la presencia de defectos en las bolas, jaula o pistas. Por otro lado, la medición de la humedad alrededor del rodamiento no es pertinente y no va a ayudar en la detección de estas fallas.

# Variables físicas a monitorear

## Ejemplo de variables físicas a medir:

Dominio	Variable física
Térmico	Temperatura, flujo de calor, disipación térmica.
Eléctrico	Voltaje, corriente, resistencia eléctrica, inductancia, impedancia, capacitancia, constante dieléctrica, carga, polarización, campo eléctrico, frecuencia, potencia, nivel de ruido.
Mecánico	Largo, área, volumen, desplazamiento, velocidad, aceleración, flujo, fuerza, densidad, densidad relativa, rigidez, fricción, presión, emisión acústica.
Químico	Concentración química, reactividad.
Humedad	Humedad relativa, humedad absoluta.
Biológico	PH, concentración de moléculas biológicas, micro-organismos.
Óptico	Intensidad lumínica, fase, largo de onda, polarización, reflectancia, transmitancia, refracción, amplitud, frecuencia.
Magnético	Campo magnético, momento magnético, permeabilidad, dirección, posición, distancia.



# Adquisición y almacenamiento

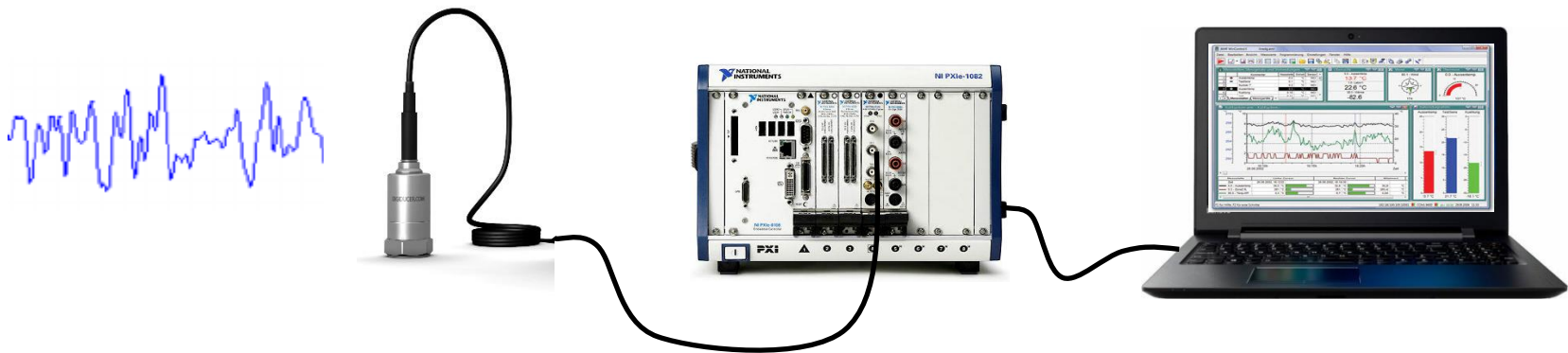
La adquisición, almacenamiento y pre-procesamiento de las señales representa el tercer paso en PHM.

Variable física

Sensor

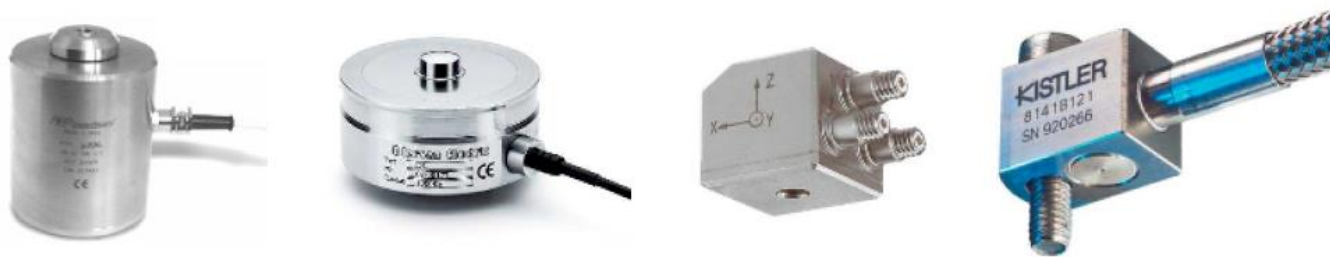
Tarjeta de adquisición

Computador



# Tipos de sensores

Para una cierta variable física a monitorear existen distintos tipos de sensores. Por ejemplo, sensores generadores de corriente, sensores de generación de carga, sensores resistivos, sensores inductivos y sensores capacitivos.



Ejemplo de sensores para medición de fuerza y aceleración

# Tipos de sensores

Las consideraciones más relevantes al momento de seleccionar un sensor son:

- Desempeño: un set de características metrológicas de los sensores (precisión, linealidad, sensibilidad, etc.).
- Confiabilidad: los sensores deben ser seleccionados de manera que no alteren la confiabilidad del sistema monitoreado.
- Costo: se deben considerar los costos para que la solución sea competitiva.
- Número y posición: el número de sensores depende de cada aplicación (redundancia, cobertura). La posición también es relevante y debe seleccionarse de manera que se puedan medir las variables deseadas.

# Tipos de sensores

Las consideraciones más relevantes al momento de seleccionar un sensor son:

- Tipo de fijación: existen diferentes soluciones para la fijación (pegamento, atornillado, magnético o con cera) y la selección depende de la calidad requerida para las mediciones y del ambiente de operación del componente.
- Dimensión y peso: se debe considerar la forma, tamaño, peso y cubierta de cada sensor de forma de no influenciar las mediciones y de respetar las restricciones de espacio.
- Ambiente: los sensores deben soportar las variaciones ambientales a las que serán sometidos (temperaturas, humedad, radiación nuclear o electromagnética, etc.)

# Sistema de adquisición

El sistema de adquisición está compuesto por la tarjeta de adquisición, un computador, un software de adquisición y procesamiento y eventualmente un disco externo para el almacenamiento de grandes volúmenes de datos.

La tarjeta de adquisición puede instalarse en el pc o venir como un sistema externo. La mayoría de las tarjetas vienen un sistema de conversión análogo-digital. Cada tarjeta viene con un software de adquisición y procesamiento. Este software permite configurar, la frecuencia de muestreo, tiempo de adquisición y otros.

# Sistema de adquisición

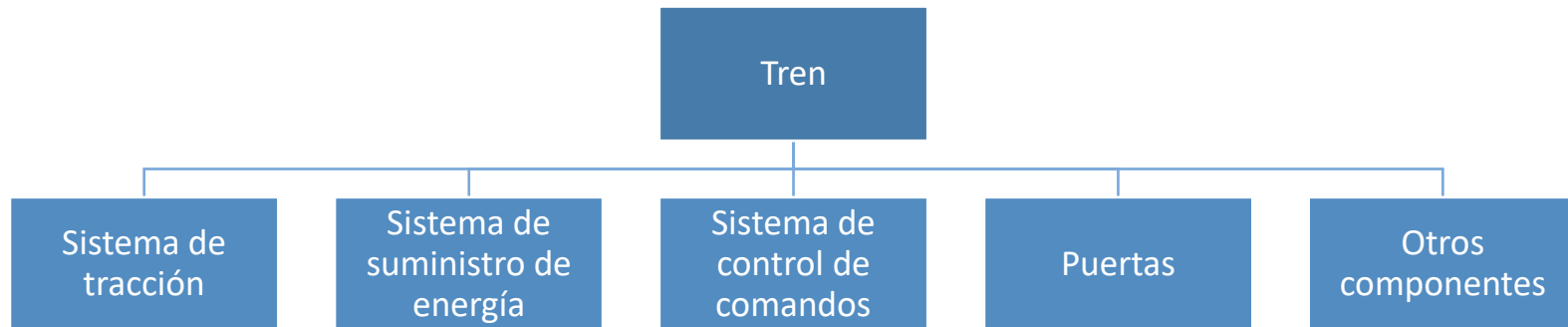
Los parámetros más importantes de una tarjeta de adquisición son: la resolución (bits), velocidad de transferencia de datos, búfer de datos, número de canales, frecuencia máxima de adquisición.

Los datos adquiridos son guardados en archivos en distintos formatos (usualmente .txt, .csv o .mat). Estos datos pueden ser leídos, analizados y procesados posteriormente por programas o algoritmos desarrollados especialmente para aplicaciones de PHM.

# Ejemplo - PHM en rodamientos

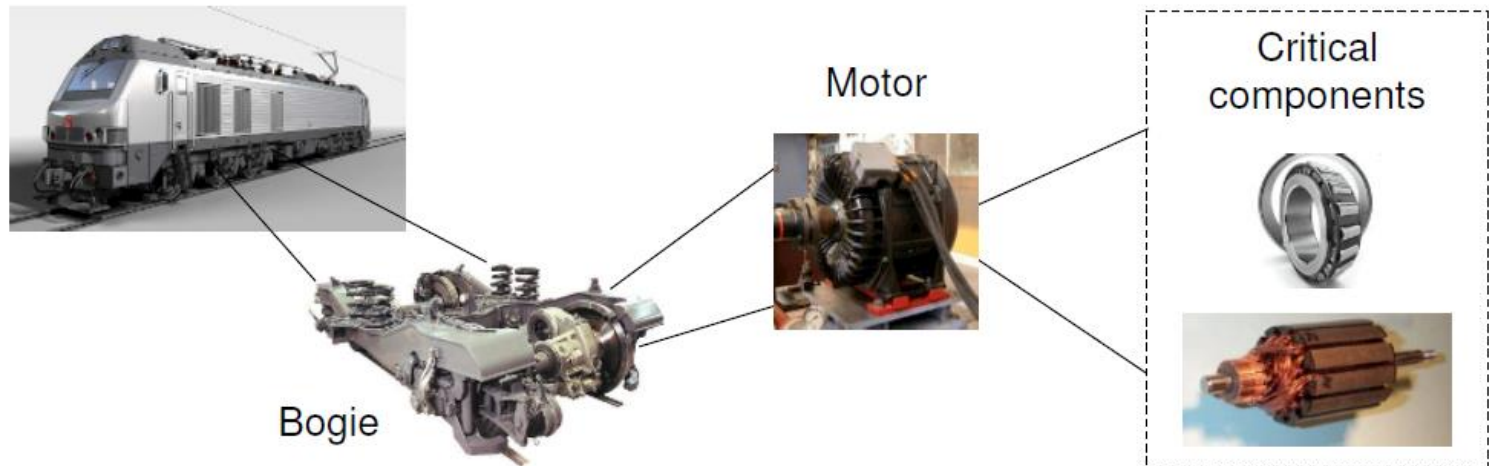
Para ilustrar el procedimiento consideremos un tren de pasajeros como caso de estudio.

El tren es un sistema complejo, compuesto por varios subsistemas.



# Ejemplo - PHM en rodamientos

El sistema de tracción en particular, esta compuesto por el bogie que lleva el motor eléctrico. Este último consta de un rotor (basado en un imán permanente) y un estator (basado en bobinas).





# Ejemplo - PHM en rodamientos

Del análisis de fallas se concluye que los rodamientos y el estator son los componentes asociados al mayor número de fallas.

Failure percentage %					
Component	Bloch & Geitner [BLO 99]	O'Donnell [O'DO 85]	IEEE-ERPI [LAN 02]	Albrecht <i>et al.</i> [ALB 86]	Alstom transport
Bearings	41	45–50	45–55	41	40
Stator	37	30–40	26–36	36	38
Axle	10	8–12		9	10
Other	12			14	12

Distribución de fallas en motores

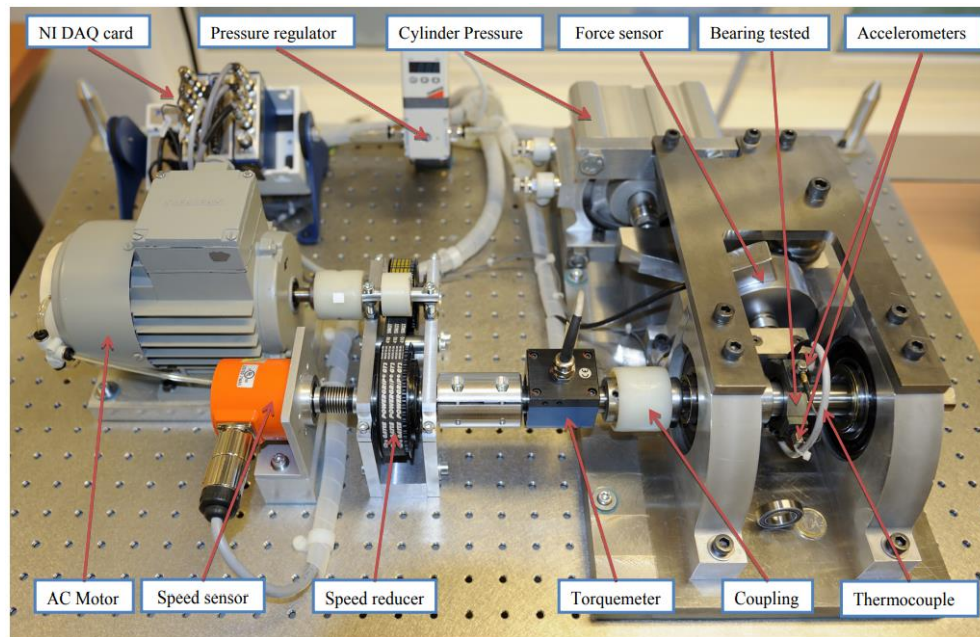
# Ejemplo - PHM en rodamientos

De un análisis de Análisis de Modos y Efectos de Falla y Criticidad (FMECA) se concluyó que los rodamientos tienen la mayor criticidad. Esto se debe al hecho que su falla es suficiente para detener el eje del motor, lo que puede llevar a una detención del tren.

Los rodamientos pueden fallar por distintos motivos relacionados a desgaste, mala lubricación o presencia de objetos extraños, entre otros. Por lo tanto, no es fácil definir a priori como son las mediciones de un rodamiento defectuoso y es necesario realizar un estudio más detallado.

# Ejemplo - PHM en rodamientos

Para generación la información se trabajó en un montaje experimental que permite operar los rodamientos en condiciones desfavorables (velocidad y carga). Obteniendo una gran cantidad de información sobre la degradación del rodamiento.



# Ejemplo - PHM en rodamientos

Se realizaron varios experimentos con diferentes velocidades de operación y carga.

FC	# Exper.	Speed (rpm)	Load (N)	Duration	Fault mode
1	1	1800	4000	3h25	IR, OR
	2			6h50	OR
	3			6h48	OR
	4			6h16	OR
	5			1h00	OR
	6			1h12	OR, IR
2	1	1650	4200	1h12	IR
	2			5h25	OR
	3			2h05	OR
	4			6h26	OR
	5			1h57	IR, OR
	6			2h11	OR

FC = Functioning Condition, OR = Outer Race, IR = Inner Race

# Ejemplo - PHM en rodamientos

Ejemplo de degradación en rodamiento.



# Ejemplo 1

```
import pandas as pd
import matplotlib.pyplot as plt

#registro de datos de vibraciones en RMS
Datos1 = pd.read_csv('RMSvibraciones1.txt')
Datos2 = pd.read_csv('RMSvibraciones2.txt')
Datos3 = pd.read_csv('RMSvibraciones3.txt')

#registro de tiempo en minutos
t1 = pd.read_csv('time1.txt')
t2 = pd.read_csv('time2.txt')
t3 = pd.read_csv('time3.txt')

plt.figure()
plt.plot(t1/60,Datos1)
plt.xlabel('tiempo (horas)', fontsize=14)
plt.ylabel('RMS vibraciones', fontsize=14)
plt.legend(['R1','R2','R3','R4'])
plt.title('Falla en pista interna R3 y falla en bolas R4')
```

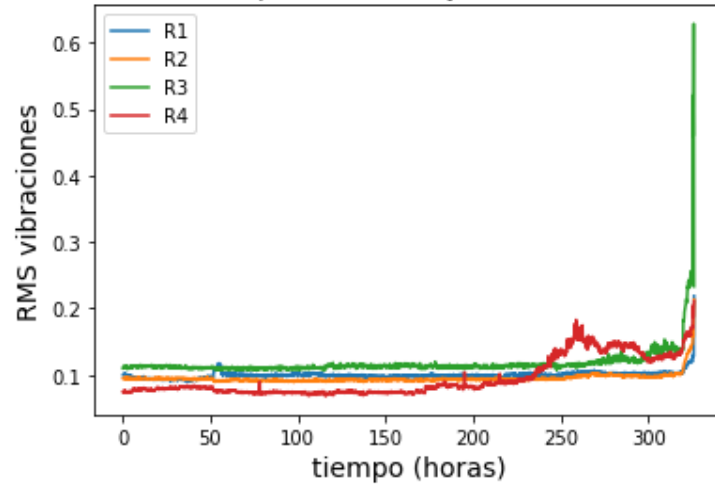
# Ejemplo 1

```
plt.figure()  
plt.plot(t2/60,Datos2)  
plt.xlabel('tiempo (horas)', fontsize=14)  
plt.ylabel('RMS vibraciones', fontsize=14)  
plt.legend(['R1','R2','R3','R4'])  
plt.title('Falla en pista externa R1')
```

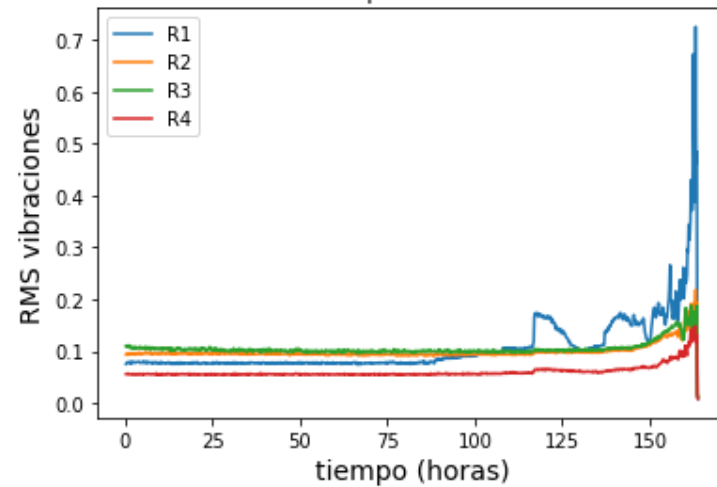
```
plt.figure()  
plt.plot(t3/60,Datos3)  
plt.xlabel('tiempo (horas)', fontsize=14)  
plt.ylabel('RMS vibraciones', fontsize=14)  
plt.legend(['R1','R2','R3','R4'])  
plt.title('Falla en pista externa R3')
```

# Ejemplo 1

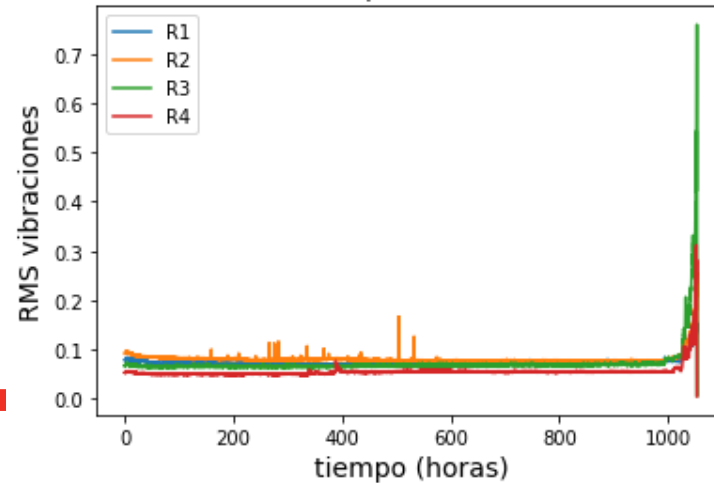
Falla en pista interna R3 y falla en bolas R4



Falla en pista externa R1



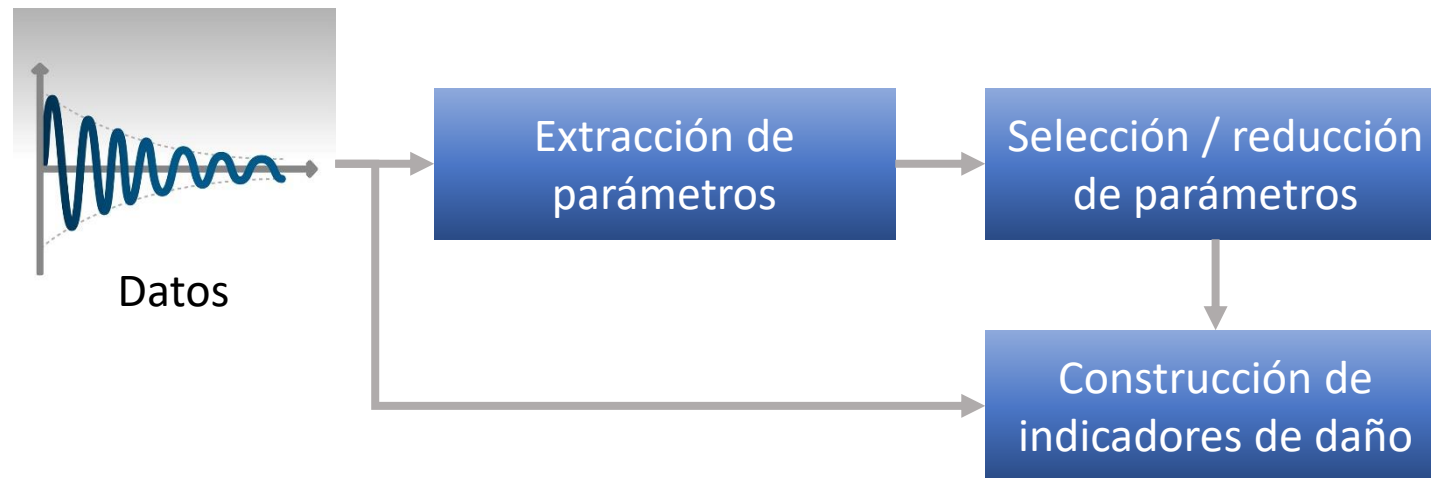
Falla en pista externa R3





# Procesamiento de los datos

Los datos adquiridos tienen información relevante sobre la aparición y evolución de una falla. Sin embargo, su uso directo no es sencillo y es necesario procesarlos.



# Procesamiento de los datos

- **Extracción de parámetros:** tiene por objetivo transformar la señal bruta en otra señal en el dominio del tiempo, frecuencia o tiempo-frecuencia, la que es luego procesada para construir indicadores que contengan información relacionada a la degradación en el sistema en estudio.

La extracción de parámetros requiere de conocimiento del fenómeno físico involucrado y además depende del uso. Los parámetros usados para detección pueden ser distintos a los usados para diagnóstico o pronóstico.

# Procesamiento de los datos

- **Selección/reducción de parámetros:** Es bastante común la extracción de más parámetros de los realmente necesarios, lo que dificulta el procesamiento. Hay dos estrategias para resolver esto:
  - Reducción de parámetros: Los parámetros se proyectan en un espacio de dimensión menor mientras que se mantiene la varianza de los datos.
  - Selección de parámetros: De forma complementaria, también es posible seleccionar un set de parámetros de acuerdo a algún criterio. Existen diferentes métricas que miden que tan adecuados son los parámetros.

# Procesamiento de los datos

- **Construcción de indicadores de daño:** Los indicadores de daño son señales construidas a partir de la señal original o de los parámetros seleccionados. En ambos casos, se requiere de procesamiento (fusión de datos, filtros, etc.) con el propósito de obtener descriptores que contengan suficiente información para revelar el estado del componente.

# Extracción de parámetros

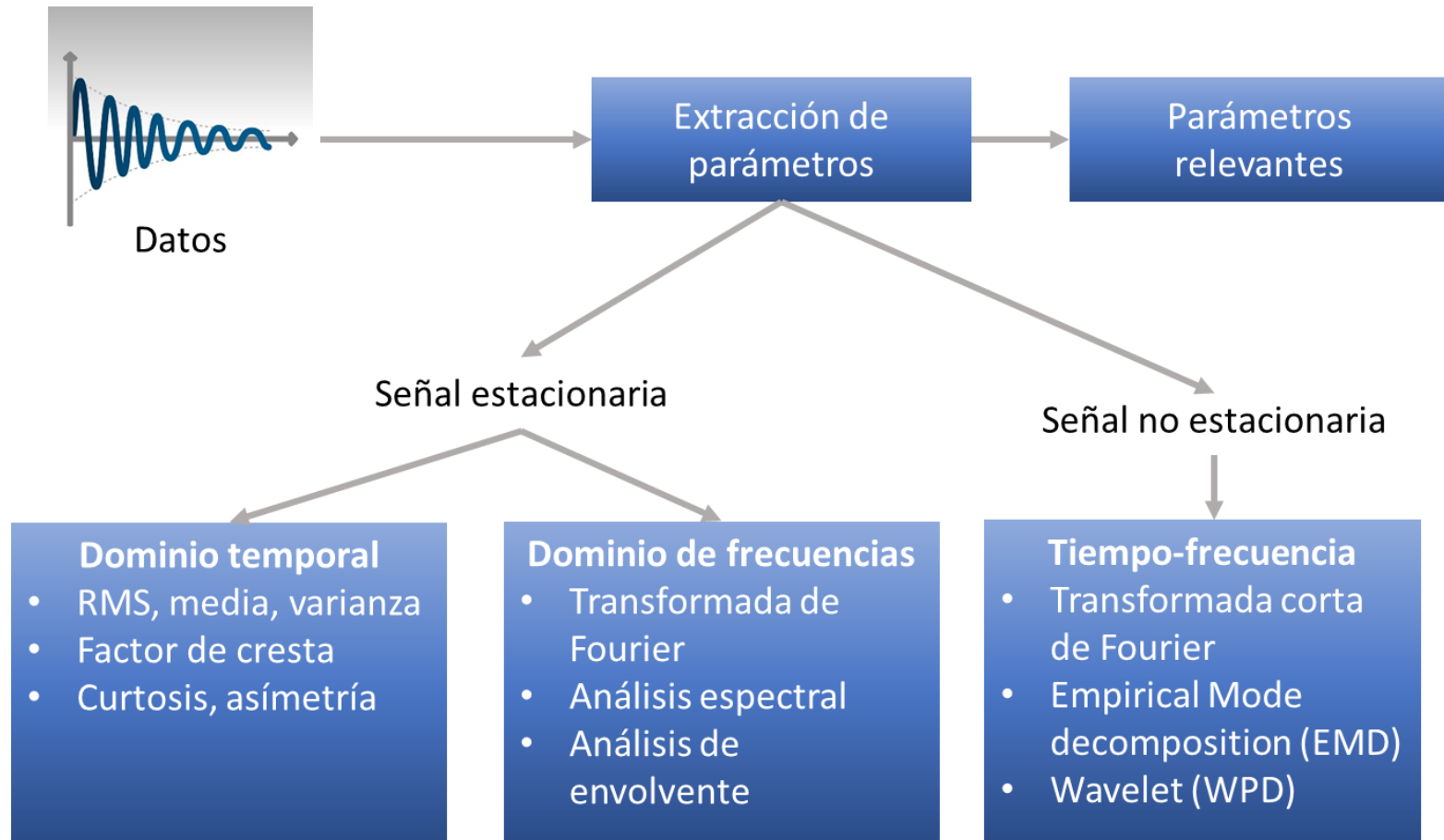
# Extracción de parámetros

La extracción de parámetros consiste en procesar los datos brutos para construir indicadores (parámetros) que se puedan interpretar o que al menos contengan suficiente información para los algoritmos de detección, diagnóstico y pronóstico.

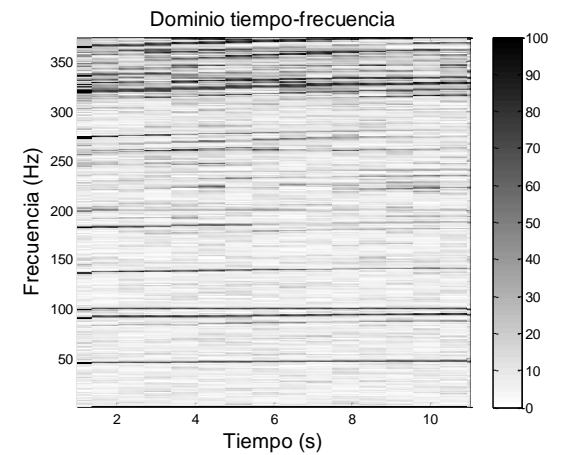
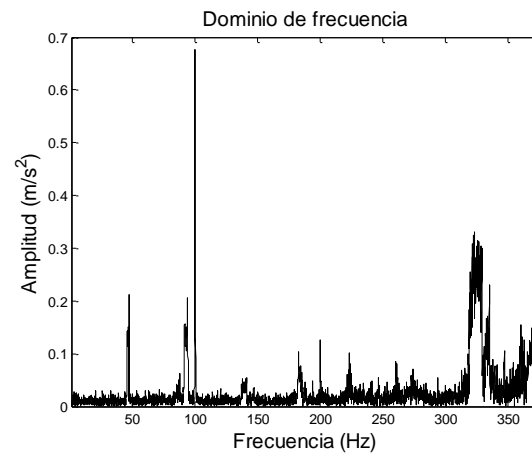
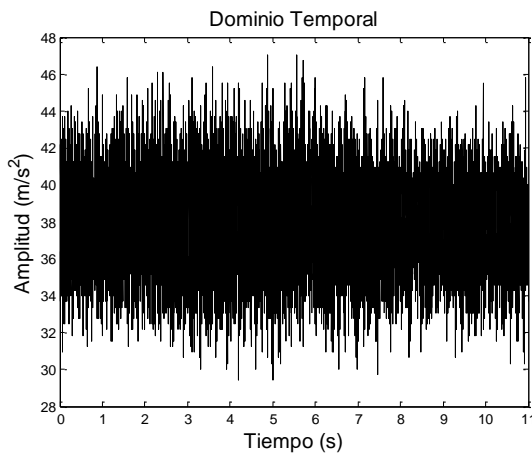
Los métodos de extracción dependen del dominio de la señal y se pueden separar en dos categorías, dependiendo si la señal es o no estacionaria.

- Una señal estacionaria, en la que sus propiedades promedio no varían en el tiempo. Se pueden utilizar métodos en el dominio temporal o de frecuencia.
- Una señal no estacionaria varía en el tiempo. Por lo tanto, se deben utilizar métodos en el dominio tiempo-frecuencia.

# Extracción de parámetros



# Extracción de parámetros





# Dominio del tiempo

Los parámetros en el dominio temporal se basan en el calculo de parámetros estadísticos de la señal.

Consideremos una señal  $x(t) = \{x_1 \quad x_2 \quad \cdots \quad x_n\}$ , se pueden calcular los siguientes parámetros:

- **RMS**, El valor RMS describe el contenido de energía de la señal. RMS se utiliza para evaluar el estado general de los componentes. Debido a esto no es muy sensible a fallo incipiente, sin embargo, se utiliza para realizar un seguimiento de la progresión de fallo general:

$$x_{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}$$

# Dominio del tiempo

- **Valor peak**, el valor peak es la máxima amplitud de la señal en cierto intervalo de tiempo.:

$$x_p = \max_i x_i$$

- **Amplitud Peak to Peak**, corresponde a la distancia entre la amplitud máxima y mínima de una señal :

$$x_{p-p} = \max_i x_i - \min_i x_i$$

- **Factor de cresta**, da cuenta de la diferencia entre el peak de una señal respecto a su valor RMS. El valor del Factor cresta es normalmente entre 2 y 6. Un factor cresta mayor a 6 indica un posible fallo de la máquina.

$$CF = \frac{x_p}{x_{RMS}}$$

# Dominio del tiempo

- **Media aritmética**, se refiere al valor promedio de la señal en un intervalo de tiempo

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

- **Varianza** ( $\sigma^2$ )(segundo momento)

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$$

# Dominio del tiempo

- **Asimetría** (tercer momento), indica la simetría en la amplitud de la función de densidad de probabilidad de una serie de tiempo. Una serie de tiempo con un número igual de amplitudes grandes y pequeñas tiene una asimetría de cero.

$$\gamma_3 = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^3}{\sigma^3} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^3}{\left( \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \right)^3}$$

# Dominio del tiempo

- **Kurtosis** (cuarto momento), es un parámetro estadístico que da cuenta de cuan agudo son en promedio los peaks de una señal. Si el valor kurtosis es cercano a 3, se tendrá una distribución Gaussiana en relación a los peaks. Kurtosis mayores a 3 implican peaks más puntiagudos y Kurtosis menores a 3 implican peaks más planos.

$$\gamma_4 = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^4}{\sigma^4} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^4}{\left( \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \right)^2}$$

# Dominio del tiempo

- Momentos centrales de orden mayor

$$\gamma_m = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^m}{\sigma^m} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^m}{\left( \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \right)^m}$$

# Dominio del tiempo

Los parámetros anteriores se pueden calcular con la señal original o con la derivada o integral de la señal.

- La derivada de la señal acentúa componentes a frecuencias altas,

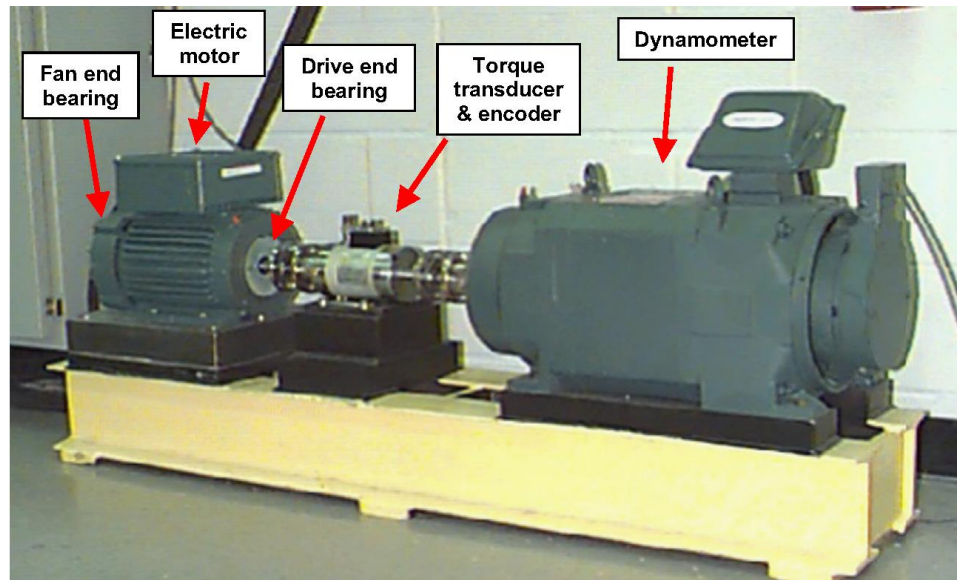
$$dx_i = x_i - x_{i-1}$$

- La integral de la señal acentúa componentes a frecuencias bajas,

$$x_i = x_i + x_{i-1}$$

# Ejemplo 2

- Se introdujeron fallas en los dos rodamientos que soportan el eje del motor
- Se midieron las vibraciones en ambos rodamientos.
- Velocidad de giro aproximado 1700 RPM





# Ejemplo 2

```
#importar librerías
import scipy.io as sio
import numpy as np
import matplotlib.pyplot as plt
import math
from scipy.stats import kurtosis, skew
from numpy import mean, sqrt, square
```

```
#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']
```

```
#vector de tiempo
Fs=48828 #sampling rate
dt=1/Fs #paso de tiempo
N=len(Normal)
t=np.linspace(0,dt*(N-1),N)
```

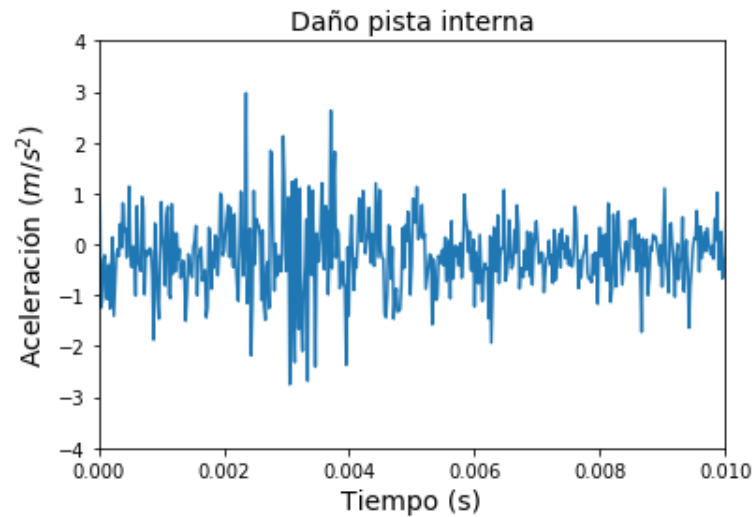
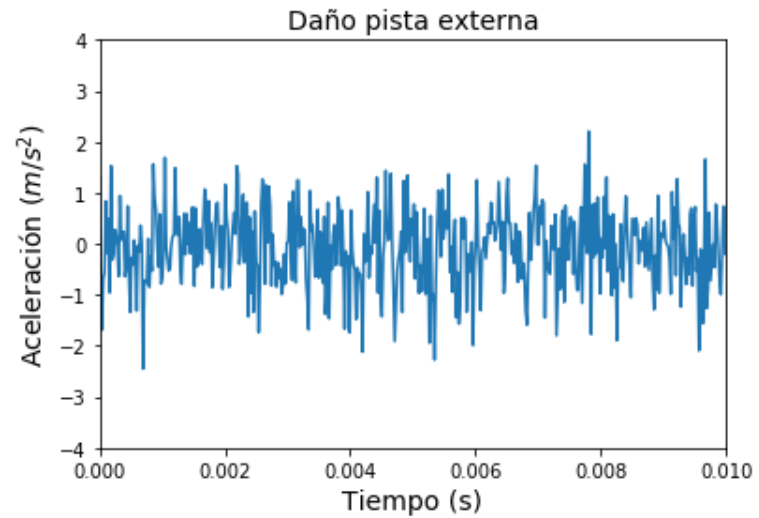
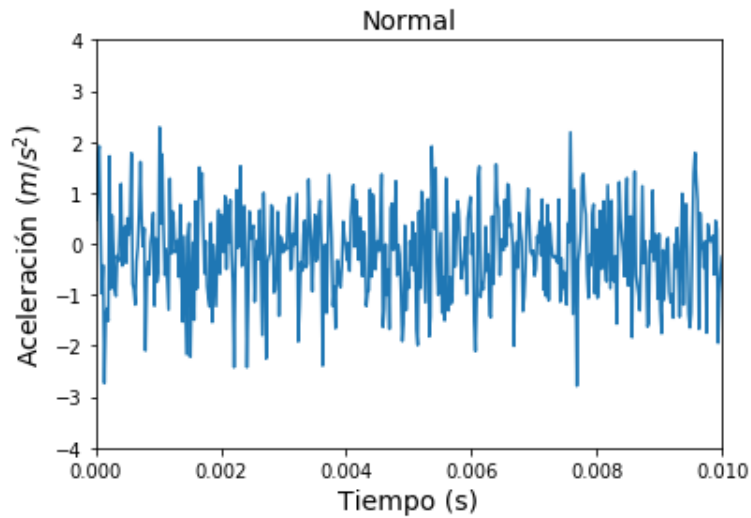
# Ejemplo 2

```
plt.plot(t,Normal)
plt.xlabel('Tiempo (s)', fontsize=14)
plt.ylabel('Aceleración ' " $(m/s^2)$ ", fontsize=14)
plt.title('Normal', fontsize=14)
plt.xlim(0,0.01)
plt.ylim(-4,4)
plt.show()
```

```
plt.plot(t,Outer)
plt.xlabel('Tiempo (s)', fontsize=14)
plt.ylabel('Aceleración ' " $(m/s^2)$ ", fontsize=14)
plt.title('Daño pista externa', fontsize=14)
plt.xlim(0,0.01)
plt.ylim(-4,4)
plt.show()
```

```
plt.plot(t,Inner)
plt.xlabel('Tiempo (s)', fontsize=14)
plt.ylabel('Aceleración ' " $(m/s^2)$ ", fontsize=14)
plt.title('Daño pista interna', fontsize=14)
plt.xlim(0,0.01)
plt.ylim(-4,4)
plt.show()
```

# Ejemplo 2



# Ejemplo 2

```
#calcular parametros por tramos  
L=5000 #largo de los segmentos  
l=1000 #overlap  
Nt=math.floor((N-l)/(L-l)) #total de tramos
```

```
#inicializar matrices con parametros  
Pn=np.zeros((Nt,8))  
Po=np.zeros((Nt,8))  
Pi=np.zeros((Nt,8))
```

# Ejemplo 2

```
for i in range(1,Nt+1):
    inicio=(i-1)*L-(i-1)*l+1
    fin=i*L-(i-1)*l

    Pn[i-1,0]=sqrt(mean(square(Normal[inicio:fin]))) #RMS
    Pn[i-1,1]=np.amax(Normal[inicio:fin]) #Peak
    Pn[i-1,2]=np.amax(Normal[inicio:fin])-np.amin(Normal[inicio:fin]) #peak-peak
    Pn[i-1,3]=Pn[i-1,1]/Pn[i-1,0] #crest
    Pn[i-1,4]=np.mean(Normal[inicio:fin]) #Media
    Pn[i-1,5]=np.var(Normal[inicio:fin]) #var
    Pn[i-1,6]=skew(Normal[inicio:fin])[0] #asimetria
    Pn[i-1,7]=kurtosis(Normal[inicio:fin])[0] #curtosis

    Po[i-1,0]=sqrt(mean(square(Outer[inicio:fin]))) #RMS
    Po[i-1,1]=np.amax(Outer[inicio:fin]) #Peak
    Po[i-1,2]=np.amax(Outer[inicio:fin])-np.amin(Outer[inicio:fin]) #peak-peak
    Po[i-1,3]=Po[i-1,1]/Po[i-1,0] #crest
    Po[i-1,4]=np.mean(Outer[inicio:fin]) #Media
    Po[i-1,5]=np.var(Outer[inicio:fin]) #var
    Po[i-1,6]=skew(Outer[inicio:fin])[0] #asimetria
    Po[i-1,7]=kurtosis(Outer[inicio:fin])[0] #curtosis

    Pi[i-1,0]=sqrt(mean(square(Inner[inicio:fin]))) #RMS
    Pi[i-1,1]=np.amax(Inner[inicio:fin]) #Peak
    Pi[i-1,2]=np.amax(Inner[inicio:fin])-np.amin(Inner[inicio:fin]) #peak-peak
    Pi[i-1,3]=Pi[i-1,1]/Pi[i-1,0] #crest
    Pi[i-1,4]=np.mean(Inner[inicio:fin]) #Media
    Pi[i-1,5]=np.var(Inner[inicio:fin]) #var
    Pi[i-1,6]=skew(Inner[inicio:fin])[0] #asimetria
    Pi[i-1,7]=kurtosis(Inner[inicio:fin])[0] #curtosis

np.save('Pn.npy', Pn)
np.save('Po.npy', Po)
np.save('Pi.npy', Pi)
```

# Ejemplo 2

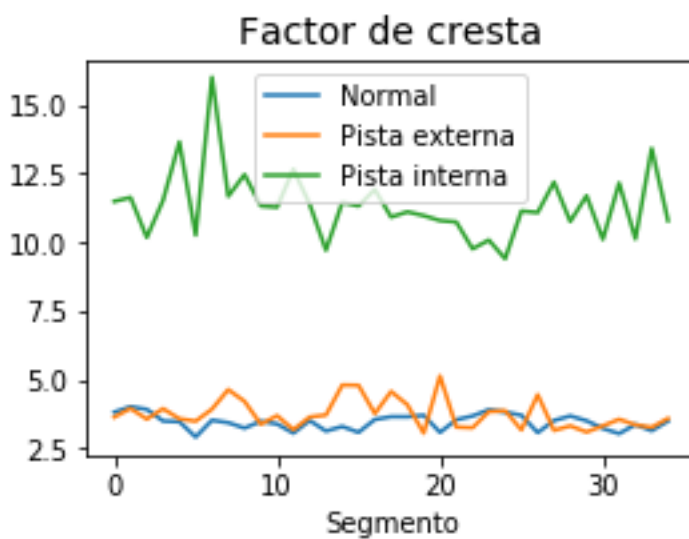
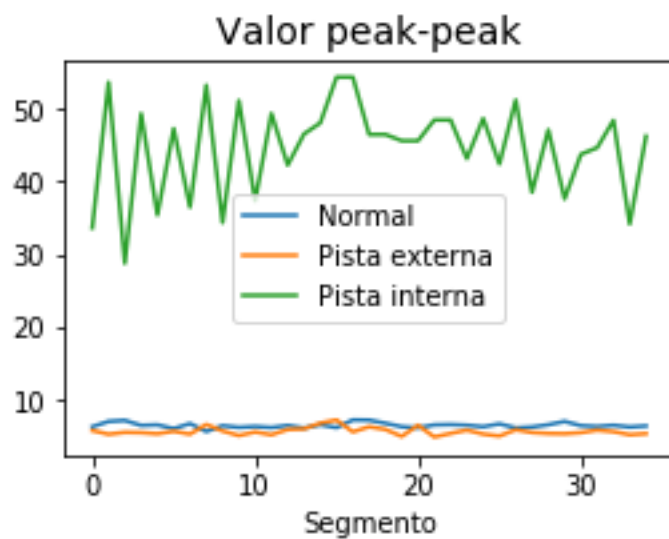
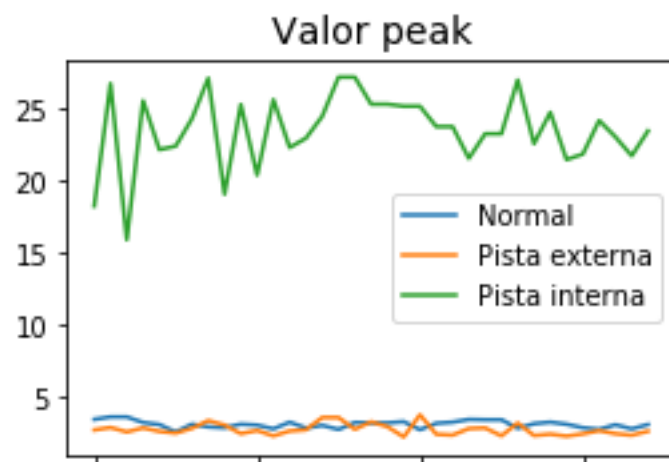
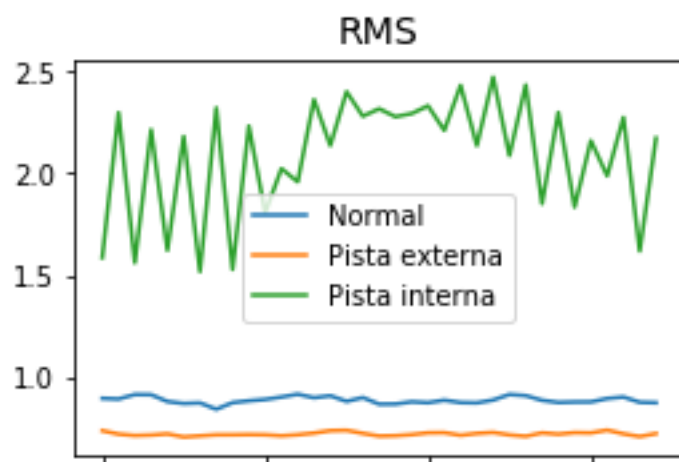
```
#graficar datos
f, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, sharex='col', figsize=(9,6))
ax1.plot(Pn[:,0])
ax1.plot(Po[:,0])
ax1.plot(Pi[:,0])
ax1.legend(['Normal', 'Pista externa', 'Pista interna'])
ax1.set_title('RMS', fontsize=14)

ax2.plot(Pn[:,1])
ax2.plot(Po[:,1])
ax2.plot(Pi[:,1])
ax2.legend(['Normal', 'Pista externa', 'Pista interna'])
ax2.set_title('Valor peak', fontsize=14)

ax3.plot(Pn[:,2])
ax3.plot(Po[:,2])
ax3.plot(Pi[:,2])
ax3.legend(['Normal', 'Pista externa', 'Pista interna'])
ax3.set_xlabel='Segmento')
ax3.set_title('Valor peak-peak', fontsize=14)

ax4.plot(Pn[:,3])
ax4.plot(Po[:,3])
ax4.plot(Pi[:,3])
ax4.legend(['Normal', 'Pista externa', 'Pista interna'])
ax4.set_xlabel='Segmento')
ax4.set_title('Factor de cresta', fontsize=14)
```

# Ejemplo 2



# Ejemplo 2

```
f, ((ax5, ax6), (ax7, ax8)) = plt.subplots(2, 2, sharex='col', figsize=(9,6))
ax5.plot(Pn[:,4])
ax5.plot(Po[:,4])
ax5.plot(Pi[:,4])
ax5.legend(['Normal', 'Pista externa', 'Pista interna'])
ax5.set_title('Media Aritmética', fontsize=14)

ax6.plot(Pn[:,5])
ax6.plot(Po[:,5])
ax6.plot(Pi[:,5])
ax6.legend(['Normal', 'Pista externa', 'Pista interna'])
ax6.set_title('Varianza', fontsize=14)

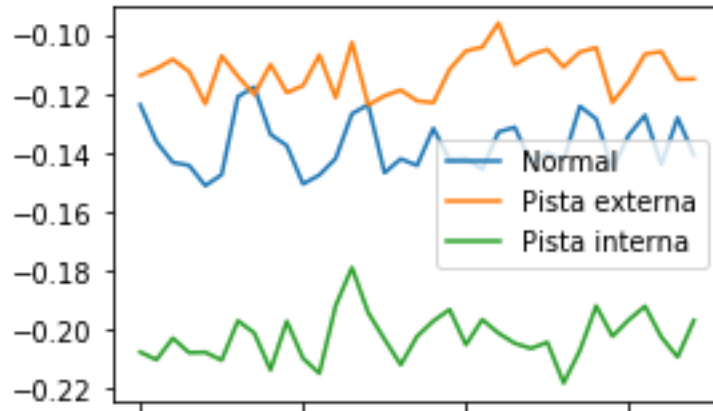
ax7.plot(Pn[:,6])
ax7.plot(Po[:,6])
ax7.plot(Pi[:,6])
ax7.legend(['Normal', 'Pista externa', 'Pista interna'])
ax7.set_xlabel('Segmento')
ax7.set_title('Asimetría', fontsize=14)

ax8.plot(Pn[:,6])
ax8.plot(Po[:,6])
ax8.plot(Pi[:,6])
ax8.legend(['Normal', 'Pista externa', 'Pista interna'])
ax8.set_xlabel('Segmento')
ax8.set_title('Curtosis', fontsize=14)
plt.show()
```

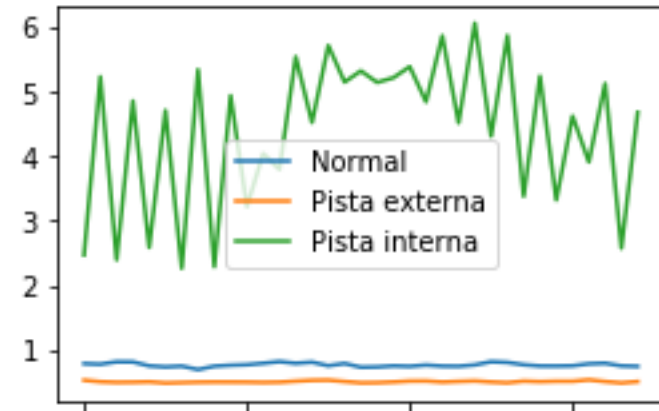


# Ejemplo 2

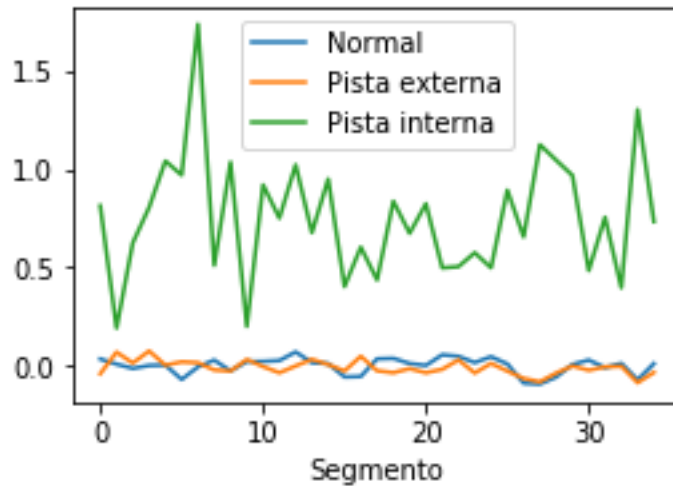
Media Aritmética



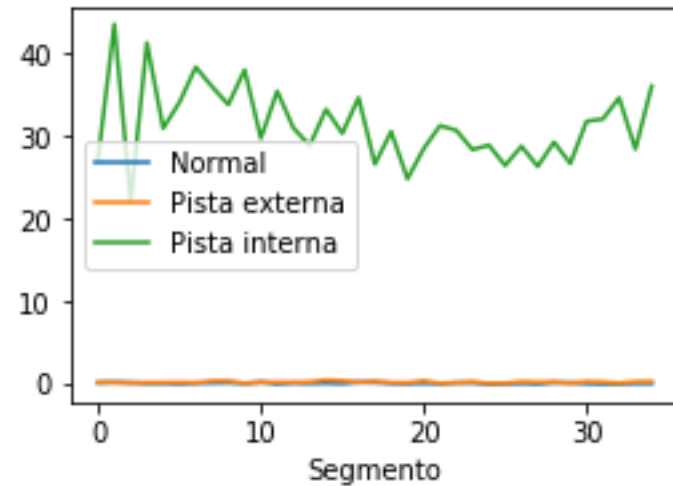
Varianza



Asimetría



Curtosis



# Ejemplo 2

Evaluar integral o derivada:

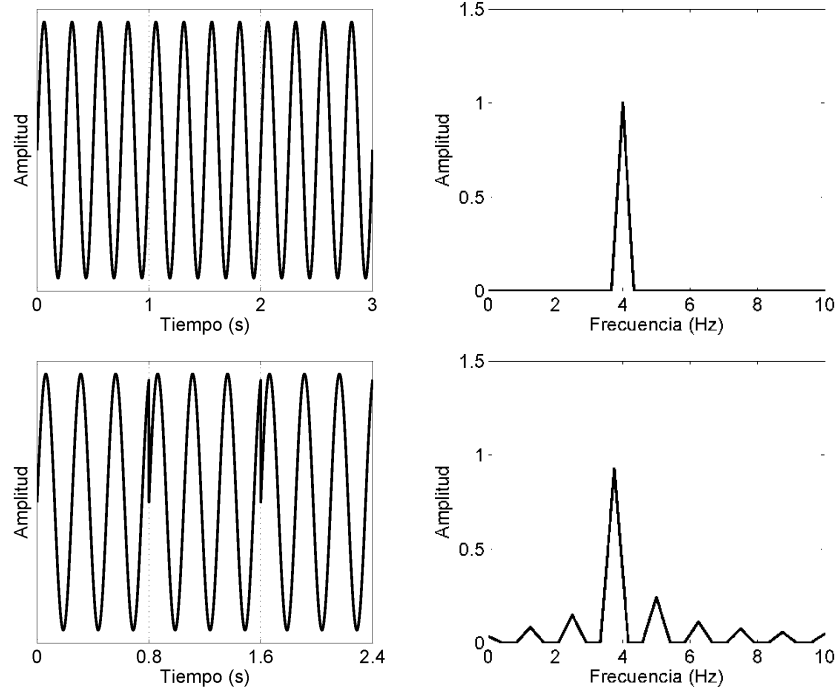
```
Normal=Normal[1:N]+Normal[0:N-1]  
Outer=Outer[1:N]+Outer[0:N-1]  
Inner=Inner[1:N]+Inner[0:N-1]  
t=np.linspace(0,dt*(N-2),N-1)
```

```
Normal=Normal[1:N]-Normal[0:N-1]  
Outer=Outer[1:N]-Outer[0:N-1]  
Inner=Inner[1:N]-Inner[0:N-1]  
t=np.linspace(0,dt*(N-2),N-1)
```

Dominio de frecuencias

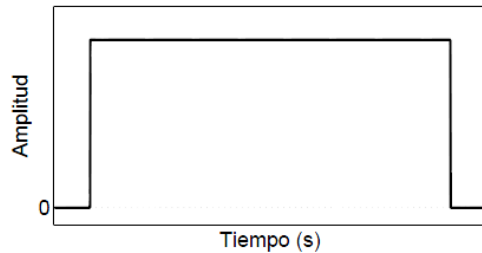
# Leakage

El Leakage se origina debido a que los datos deben ser adquiridos en un periodo de observación finito  $T$ . La transformada discreta de Fourier asume entonces que la señal es periódica con periodo  $T$ . Si esta condición no se cumple, se produce un error de “leakage”.

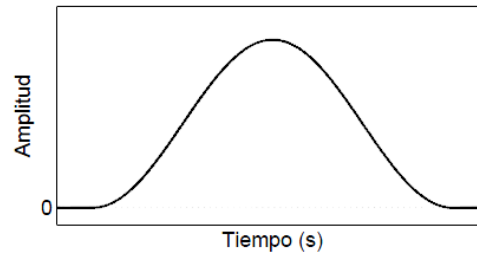


# Ventanas

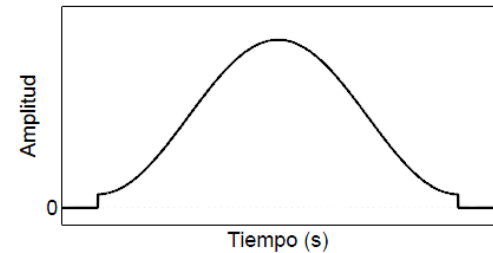
El uso de ventanas puede reducir considerablemente el error debido a leakage. En general, se buscan ventanas que reduzcan las discontinuidades en los extremos de la señal, dado que reducen el error por leakage al forzar la señal a ser periódica.



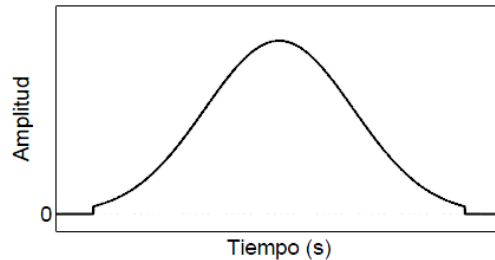
(a) Rectangular



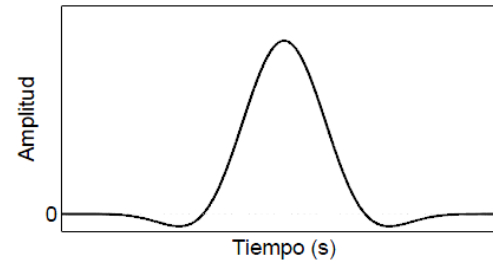
(b) Hanning



(c) Hamming

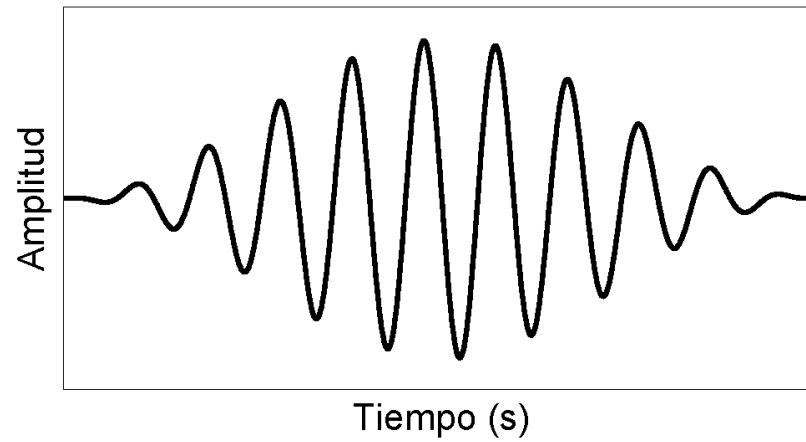
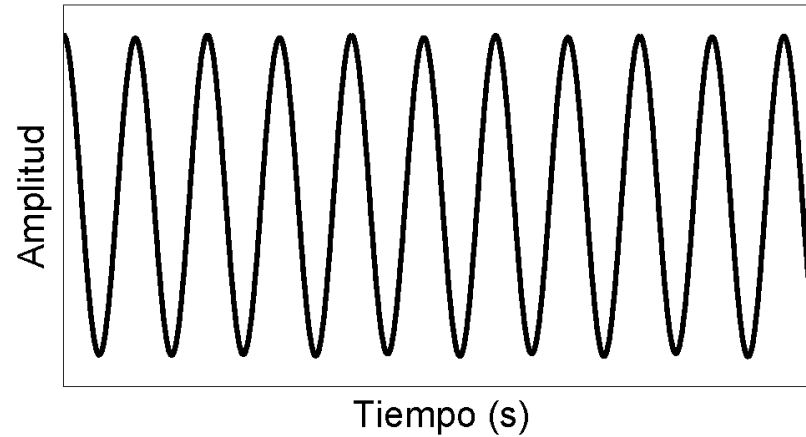


(d) Gaussian



(e) Flat-top

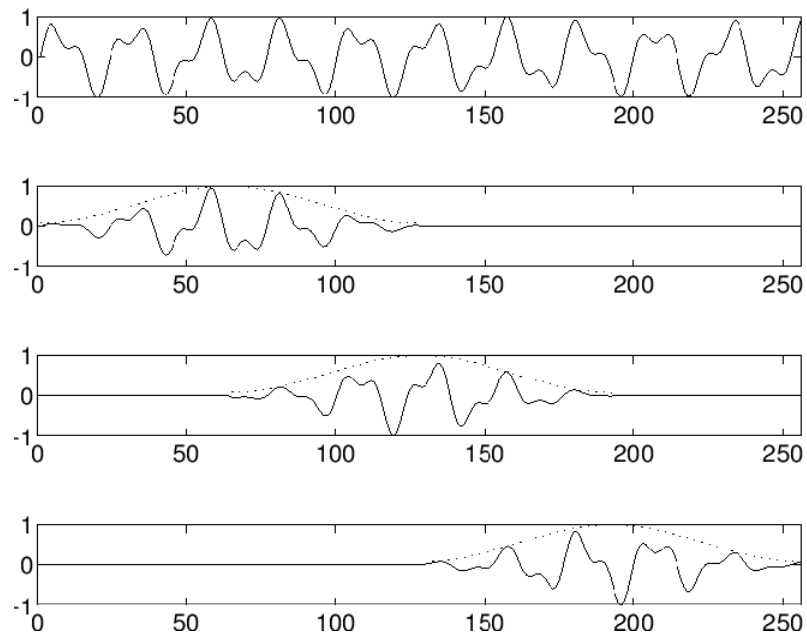
# Ventanas



# Promedios y superposición

Al utilizar solo un intervalo de tiempo para obtener un espectro, pueden aparecer componentes asociados a fuentes aleatorias de vibración (ruido experimental). Para disminuir el efecto del ruido, la señal se divide en varios intervalos, se calcula el espectro de cada intervalo y se obtiene el espectro promedio.

Estos intervalos pueden tener un grado de superposición entre ellos. Esto permite para un cierto tiempo aumentar el número de promedios.



# Ejemplo 3

```
#importar librerias
import scipy.io as sio
import numpy as np
import matplotlib.pyplot as plt
import math
from scipy.fftpack import fft, fftfreq
from numpy import mean, sqrt, square

#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 #paso de tiempo
N=len(Normal) #numero de datos

#Dividir por segmentos
L=2000 #largo de los segmentos
l=1000 #overlap
Nt=math.floor((N-l)/(L-l)) #total de segmentos
```



# Ejemplo 3

```
Pn=np.zeros((L//2-1,))
```

```
Po=np.zeros((L//2-1,))
```

```
Pi=np.zeros((L//2-1,))
```

```
ventana=np.hanning(L-1) #otras ventanas: hamming(L-1), blackman(L-1),bartlett(L-1), kaiser(L-1, beta)
```

```
for i in range(1,Nt+1):
```

```
    inicio=(i-1)*L-(i-1)*l+1
```

```
    fin=i*L-(i-1)*l
```

```
    Fn = fft(Normal[inicio:fin,0]*ventana)[1:int(L/2)]/(L/2)
```

```
    Fo = fft(Outer[inicio:fin,0]*ventana)[1:int(L/2)]/(L/2)
```

```
    Fi = fft(Inner[inicio:fin,0]*ventana)[1:int(L/2)]/(L/2)
```

```
    frq = fftfreq(L,dt)[1:int(L/2)]
```

```
Pn=Pn+abs(Fn)/Nt
```

```
Po=Po+abs(Fo)/Nt
```

```
Pi=Pi+abs(Fi)/Nt
```

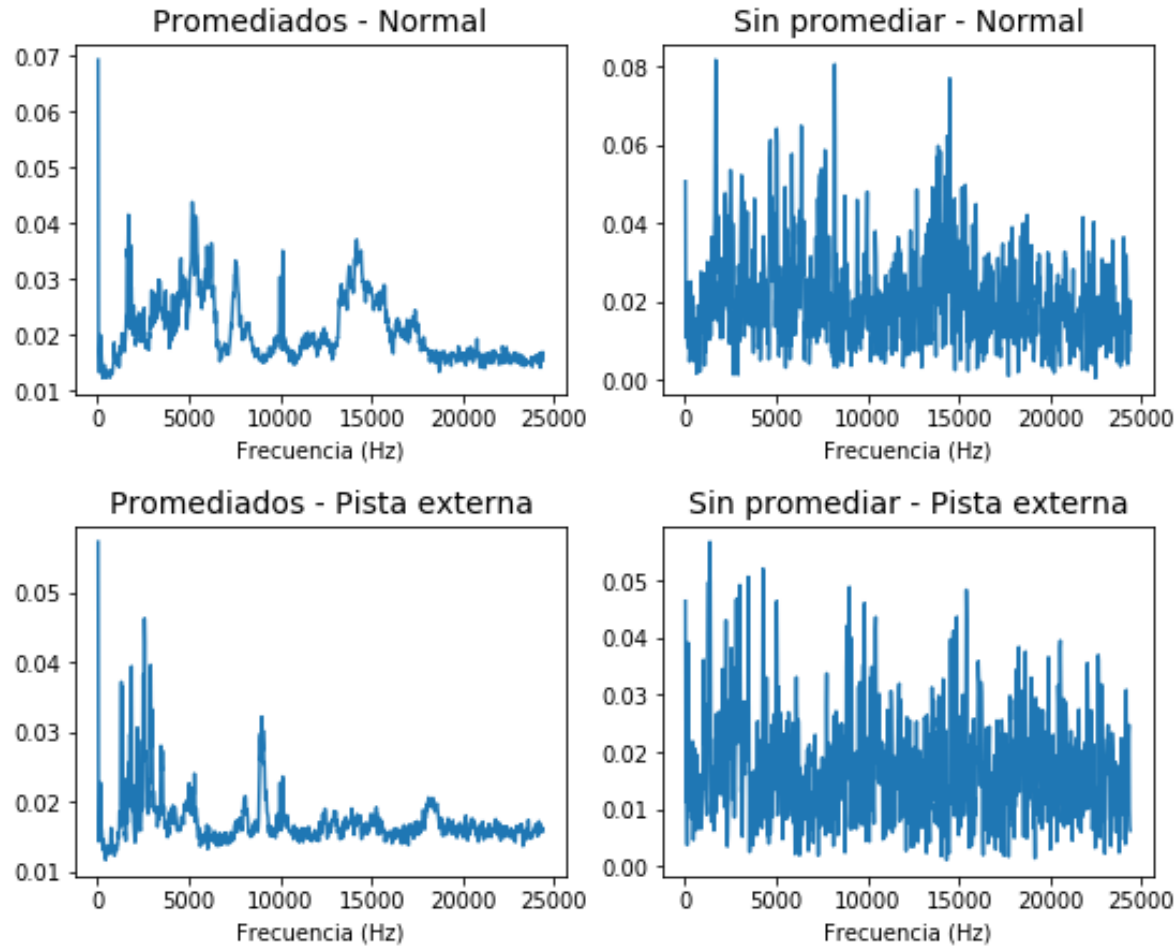
# Ejemplo 3

```
f, ((ax1, ax2)) = plt.subplots(1, 2, sharex='col', figsize=(9,3))
ax1.plot(frq,Pn)
ax1.set_xlabel='Frecuencia (Hz)'
ax1.set_title('Promediados - Normal', fontsize=14)
ax2.plot(frq,abs(Fn))
ax2.set_xlabel='Frecuencia (Hz)'
ax2.set_title('Sin promediar - Normal', fontsize=14)
plt.show()
```

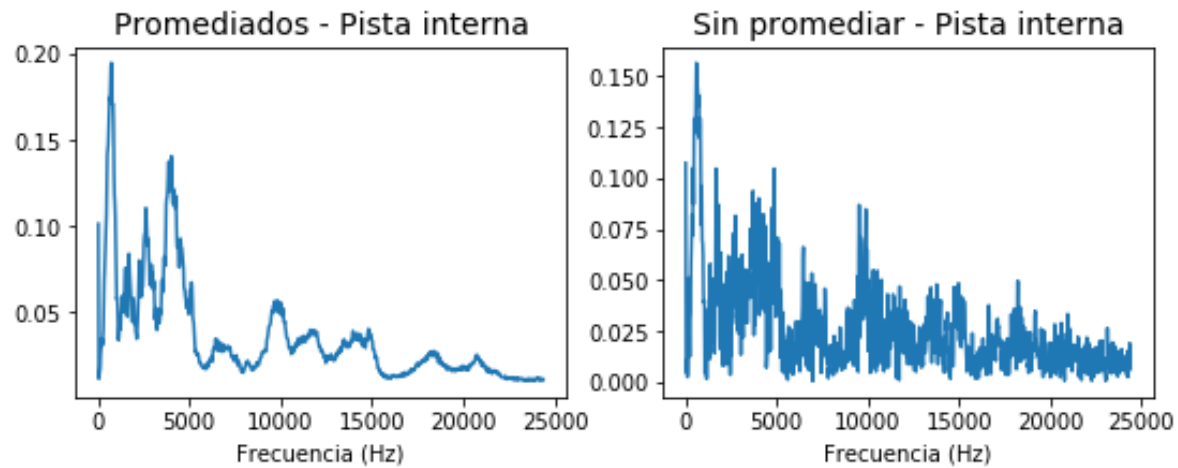
```
f, ((ax3, ax4)) = plt.subplots(1, 2, sharex='col', figsize=(9,3))
ax3.plot(frq,Po)
ax3.set_xlabel='Frecuencia (Hz)'
ax3.set_title('Promediados - Pista externa', fontsize=14)
ax4.plot(frq,abs(Fo))
ax4.set_xlabel='Frecuencia (Hz)'
ax4.set_title('Sin promediar - Pista externa', fontsize=14)
plt.show()
```

```
f, ((ax5, ax6)) = plt.subplots(1, 2, sharex='col', figsize=(9,3))
ax5.plot(frq,Pi)
ax5.set_xlabel='Frecuencia (Hz)'
ax5.set_title('Promediados - Pista interna', fontsize=14)
ax6.plot(frq,abs(Fi))
ax6.set_xlabel='Frecuencia (Hz)'
ax6.set_title('Sin promediar - Pista interna', fontsize=14)
plt.show()
```

# Ejemplo 3



# Ejemplo 3



99

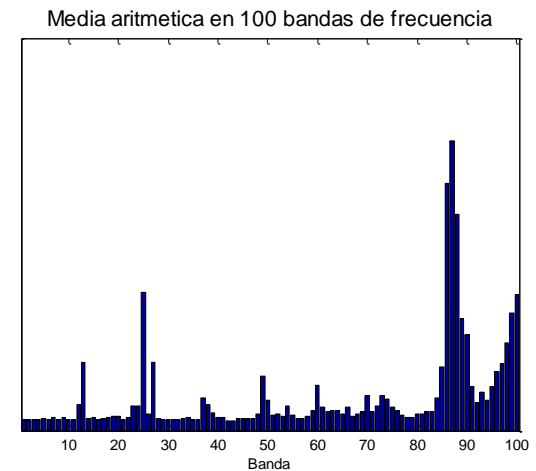
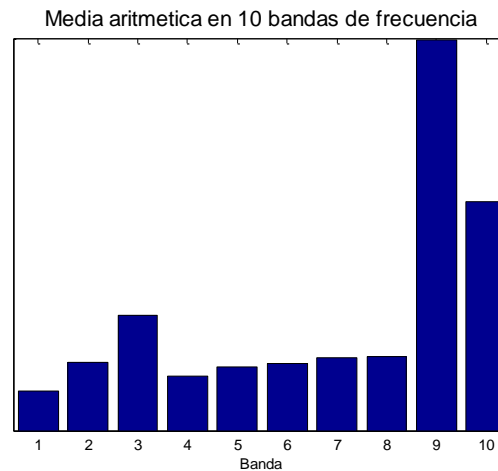
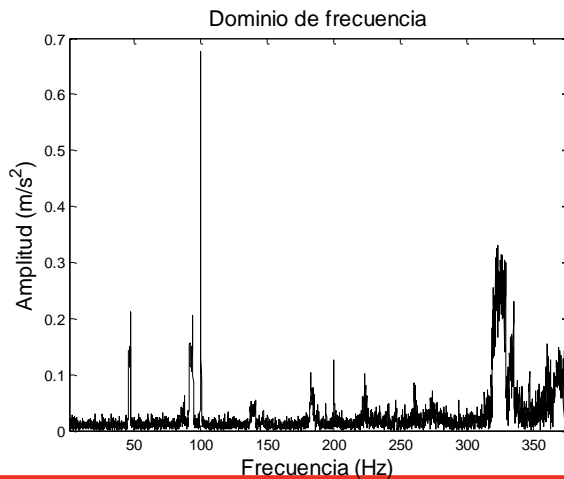


# Extracción de parámetros

$F(\omega) = \{F_1 \ F_1 \ \cdots \ F_N\}$  es la transformada de Fourier de la señal  $y(t)$ . La **media aritmética** del espectro de frecuencias dividida en bandas de frecuencias (usualmente octavos) se define como:

$$B_i = \frac{1}{N_{i+1} - N_i} \sum_{j=N_i}^{N_{i+1}-1} \text{abs}(F_j)$$

donde  $N_i$  y  $N_{i+1}$  son el primer y último componente de la banda  $i$ .



# Ejemplo 4

```
#importar librerias
import scipy.io as sio
import numpy as np
import matplotlib.pyplot as plt
import math
from scipy.fftpack import fft, fftfreq
from numpy import mean, sqrt, square

#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 #paso de tiempo
N=len(Normal) #numero de datos

#Dividir por segmentos
L=5000 #largo de los segmentos
l=1000 #overlap
Nt=math.floor((N-l)/(L-l)) #total de segmentos
nb=8 #numero de bandas
```

# Ejemplo 4

```
En=np.zeros((Nt,nb))
Eo=np.zeros((Nt,nb))
Ei=np.zeros((Nt,nb))

for i in range(1,Nt+1):
    inicio=(i-1)*L-(i-1)*l+1
    fin=i*L-(i-1)*l

    Fn = fft(Normal[inicio:fin,0])[0:int(L/2)]/(L/2)
    Fo = fft(Outer[inicio:fin,0])[0:int(L/2)]/(L/2)
    Fi = fft(Inner[inicio:fin,0])[0:int(L/2)]/(L/2)
    frq = fftfreq(L,dt)[0:int(L/2)]

    Lb=int(L/2/nb)
    for k in range(1,nb+1):
        inicio=Lb*(k-1)+1
        fin=k*Lb
        En[i-1][k-1]=mean(abs(Fn[inicio:fin]))
        Eo[i-1][k-1]=mean(abs(Fo[inicio:fin]))
        Ei[i-1][k-1]=mean(abs(Fi[inicio:fin]))
```

# Ejemplo 4

```
f, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, sharex='col', figsize=(9,6))
```

```
ax1.plot(En[:,0])  
ax1.plot(Eo[:,0])  
ax1.plot(Ei[:,0])  
ax1.legend(['Normal', 'Pista externa', 'Pista interna'])  
ax1.set_title('Banda 1', fontsize=14)
```

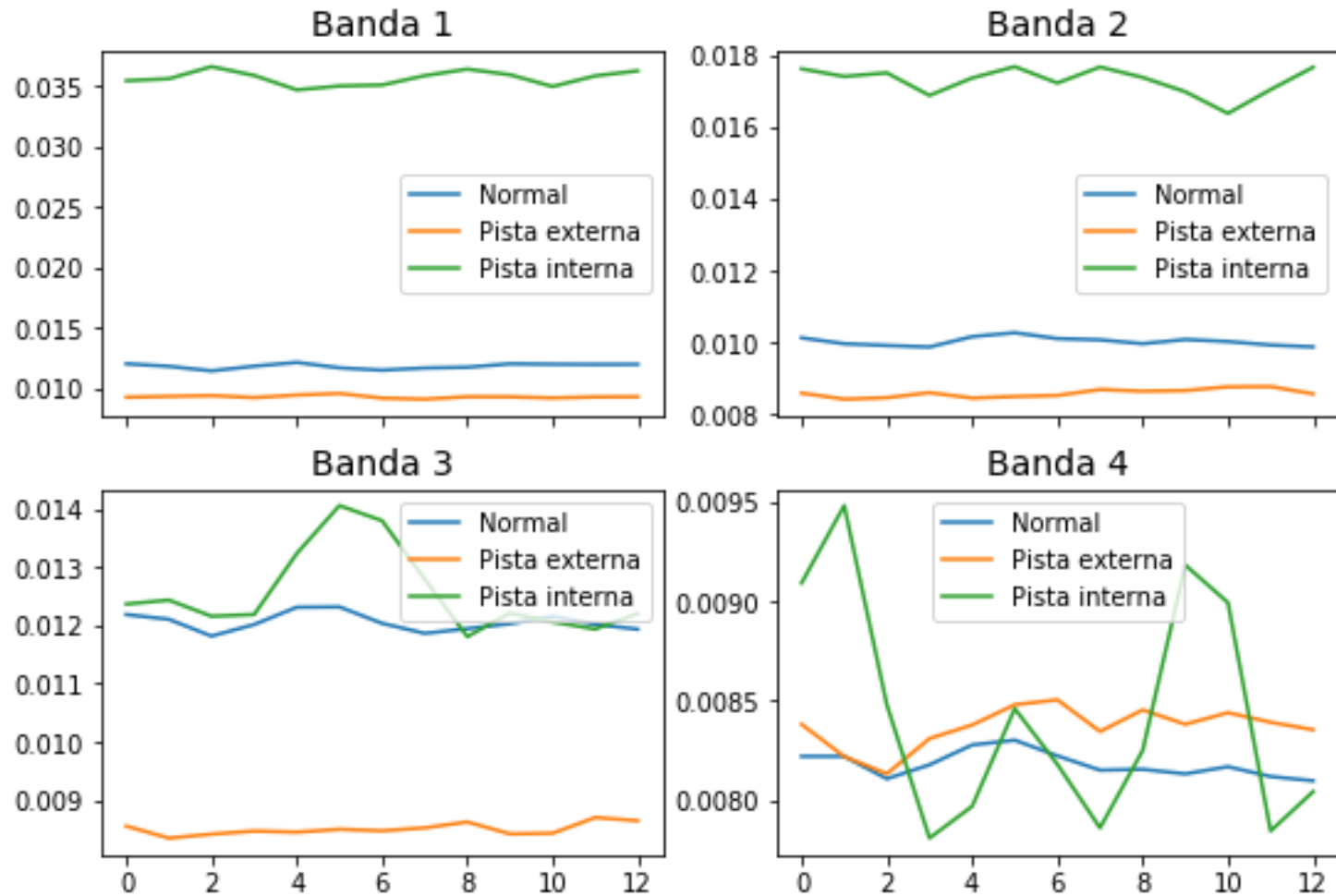
```
ax2.plot(En[:,1])  
ax2.plot(Eo[:,1])  
ax2.plot(Ei[:,1])  
ax2.legend(['Normal', 'Pista externa', 'Pista interna'])  
ax2.set_title('Banda 2', fontsize=14)
```

```
ax3.plot(En[:,2])  
ax3.plot(Eo[:,2])  
ax3.plot(Ei[:,2])  
ax3.legend(['Normal', 'Pista externa', 'Pista interna'])  
ax3.set_title('Banda 3', fontsize=14)
```

```
ax4.plot(En[:,3])  
ax4.plot(Eo[:,3])  
ax4.plot(Ei[:,3])  
ax4.legend(['Normal', 'Pista externa', 'Pista interna'])  
ax4.set_title('Banda 4', fontsize=14)
```



# Ejemplo 4



# Ejemplo 4

```
f, ((ax5, ax6), (ax7, ax8)) = plt.subplots(2, 2, sharex='col', figsize=(9,6))
ax5.plot(En[:,4])
ax5.plot(Eo[:,4])
ax5.plot(Ei[:,4])
ax5.legend(['Normal', 'Pista externa', 'Pista interna'])
ax5.set_title('Banda 5', fontsize=14)

ax6.plot(En[:,5])
ax6.plot(Eo[:,5])
ax6.plot(Ei[:,5])
ax6.legend(['Normal', 'Pista externa', 'Pista interna'])
ax6.set_title('Banda 6', fontsize=14)

ax7.plot(En[:,6])
ax7.plot(Eo[:,6])
ax7.plot(Ei[:,6])
ax7.legend(['Normal', 'Pista externa', 'Pista interna'])
ax7.set_title('Banda 7', fontsize=14)

ax8.plot(En[:,7])
ax8.plot(Eo[:,7])
ax8.plot(Ei[:,7])
ax8.legend(['Normal', 'Pista externa', 'Pista interna'])
ax8.set_title('Banda 8', fontsize=14)

np.save('En.npy', En)
np.save('Eo.npy', Eo)
np.save('Ei.npy', Ei)
```

# Ejemplo 4

