



UNIVERSIDAD
DE CANTABRIA

MEMORIA FINAL DE PRÁCTICAS

PROCESAMIENTO DE SEÑALES: ANÁLISIS ESPECTRAL SIGNAL PROCESSING: SPECTRAL ANALYSIS

Julio/Agosto 2020



Irene Goñi Ibaceta

Tutora académica: Cecilia Pola Méndez
Tutor en la empresa: David Vilasack Vilasack

Índice

1. Aspectos generales	2
1.1. Datos del alumno	2
1.2. Datos de las prácticas:	2
1.3. Resumen\Abstract	2
2. Introducción	3
2.1. Descripción del proyecto	3
2.2. Entorno de trabajo y lenguajes	4
2.3. Datasets	4
2.4. Paquetes necesarios en Python	4
2.5. Herramientas y funciones utilizadas en Matlab	5
3. Análisis en el dominio de tiempo	6
3.1. Importación de datos y almacenaje	6
3.2. Correlación	7
3.3. Representación	8
3.4. Relación entre los canales: índice de Jaccard	10
3.5. Estadísticos principales	11
4. Análisis en el dominio de la frecuencia	12
4.1. Transformada rápida de Fourier	12
4.2. Estadísticos en el espectro	14
4.3. Filtrado de señales para eliminar fluctuaciones	14
4.3.1. Ventanas	14
4.3.2. Cepstrum	15
4.3.3. Otros filtros: paso alto/bajo	17
4.4. Descomposición de la señal: Variational Mode Decomposition (VMD)	18
4.5. Kurtosis espectral y kurtograma	18
4.6. Envolvente espectral	20
5. GitHub	21
6. Valoración final	23

1. Aspectos generales

1.1. Datos del alumno

1. Nombre: Irene Goñi Ibaceta.
2. Titulación: Grado en Matemáticas.
3. Curso: 2019-2020
4. Fecha de presentación: 1 de septiembre de 2020.

1.2. Datos de las prácticas:

1. Período de realización: del 01 de julio de 2020 al 31 de agosto de 2020.
2. Entidad colaboradora: CIC (Consulting Informático de Cantabria).
Departamento: IDbox (I+D)
3. Ubicación: Santander (Cantabria).
4. Tutor de la empresa: David Vilasack Vilasack
5. Tutor académico: Cecilia Pola Méndez.

1.3. Resumen\Abstract

El objetivo principal de este proyecto es realizar un análisis, tanto en dominio de tiempo como en dominio de frecuencia, de distintas señales de rodamientos en distintos estados para, una vez aplicadas distintas técnicas y haber escogido las más rápidas y eficaces, poder tener un criterio de clasificación de estas señales. Comenzamos describiendo el software necesario para el análisis y los datos que hemos utilizado. Dedicamos la sección 3 al análisis en dominio de tiempo y la sección 4 al análisis en dominio de frecuencia. Nos centramos en poder clasificar los rodamientos en: sano, fallo de carrera interna y fallo de carrera externa. El análisis de los distintos conjuntos de datos está basado principalmente en el estudio del espectro de la señal por lo que es de gran interés la Transformada de Fourier y sus propiedades. Dedicamos la sección 5 a la exposición de un repositorio en el que pueden encontrarse las distintas funciones que han sido implementadas para el análisis de los distintos conjuntos de datos. Finalizamos el informe con una sección dedicada a la valoración de las prácticas.

The main objective of this project is to analyze several signal of different bearings with different faults, both in time and frequency domain. After the evaluation of different techniques, we will choose the fastest and most efficient ones with the aim of developing a classification criteria for these signals. We start describing the software needed for the analysis and the datasets we have used. We dedicate Section 3 to the time domain analysis and Section 4 to frequency domain analysis. Especially, we will focus on the classification of bearings in healthy, inner race fault and outer race fault. The analysis of the datasets will be mainly based on the spectrum analysis. Therefore, the Fourier Transform and its properties will be of great interest. We dedicate Section 5 to the presentation of a repository in which we can find the different functions implemented for the analysis. We end this report with a section dedicated to the assessment of the internship.

2. Introducción

2.1. Descripción del proyecto

Los rodamientos son elementos mecánicos presentes en la mayoría de las máquinas rotativas. Su vida útil depende de distintos factores: la velocidad de trabajo, la carga, la lubricación, la temperatura o de las fuerzas exteriores causadas por desalineaciones, desequilibrios, etc., por lo que es prácticamente imposible determinar su duración por métodos analíticos. La importancia de estos elementos hace necesaria la utilización de técnicas modernas de mantenimiento predictivo (basadas en el análisis de vibraciones) que contribuyan a un mejor reconocimiento de su estado.

El análisis espectral y de onda en el tiempo es determinante a la hora de diagnosticar un deterioro de un rodamiento permitiéndonos llegar a predecir el grado de deterioro y planificar su cambio con la suficiente antelación. Los rodamientos están formados por varios componentes: pista (o carrera) interior, bolas o rodillos, jaula y pista exterior. El deterioro de cada uno de estos elementos generará una o varias frecuencias características en el espectro de la señal, denominadas armónicos, que nos permitirán identificar su correspondiente fallo. Los armónicos que han sido utilizados para analizar el deterioro de un rodamiento son:

1. BPFO (Ball Pass Frequency Outer, frecuencia de deterioro de la pista exterior): número de bolas o rodillos que pasan por un punto de la carrera exterior en un giro completo.
2. BPFI (Ball Pass Frequency Inner o frecuencia de deterioro de la pista interior): número de bolas o rodillos que pasan por un punto determinado de la pista interior cada vez que el eje realiza un giro.

Estos armónicos pueden obtenerse mediante las siguientes fórmulas:

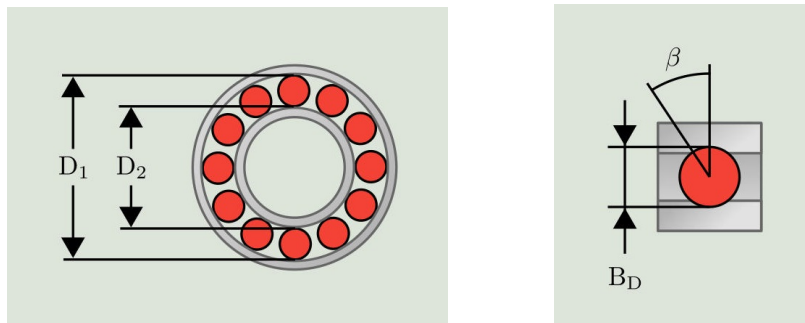
$$BPFO = \frac{N_B}{2} \cdot \left(1 - \frac{B_D}{P_D} \cos(\beta) \right) \cdot fr \quad (1)$$

$$BPFI = \frac{N_B}{2} \cdot \left(1 + \frac{B_D}{P_D} \cos(\beta) \right) \cdot fr \quad (2)$$

donde:

$$P_D = \frac{D_1 + D_2}{2}, \quad N_B = \text{Número de bolas}, \quad \beta = \text{Ángulo de contacto} \quad (3)$$

y fr es la frecuencia de giro del rodamiento.



Parámetros de los rodamientos

2.2. Entorno de trabajo y lenguajes

Se han utilizado principalmente:

1. Python: he realizado durante los primeros 10 días un curso de Python que puede encontrarse en [2]. Se ha trabajado principalmente utilizando Jupyter Notebook, un entorno informático interactivo basado en la web para crear documentos.
2. Matlab: gracias al acceso a múltiples Toolbox de Matlab, se ha utilizado este programa para la realización de algunas pruebas con algunos datasets para después migrar estas pruebas a Python (ver [5]-[7]).
3. GitHub: se ha realizado un pequeño repositorio en el que se incluyen algunos de los datasets y distintas funciones con los resultados obtenidos (ver [1]).

2.3. Datasets

El conjunto de datos usado puede encontrarse en [3]. Se realiza un análisis de diferentes señales, las cuales se clasifican en: rodamiento sano (H), con fallo en carrera interna (I) y con fallo en carrera externa (O). Otros datos relevantes del artículo son:

Tipo de rodamiento	P_D	Diámetro de bola	N_B	BPFI	BPFO
ER16K	38.52mm	7.94mm	9	5.43fr	3.57fr

Cuadro 1: Parámetros de los rodamientos

En cada conjunto de datos encontramos dos mediciones distintas: la primera de ellas, guardada en *Canal 1*, almacena la vibración de los datos medidos por el acelerómetro y la segunda, guardada en *Canal 2*, almacena la velocidad rotacional. Estas mediciones se han realizado con una frecuencia de muestreo de 200.000 Hz durante 10 segundos. En nuestro caso la frecuencia de giro es 1024rpm.

2.4. Paquetes necesarios en Python

Se han utilizado los siguientes paquetes:

scipy	pandas	matplotlib.pyplot	numpy	vmdpy
os	glob	logging	obspy.core	

Principales paquetes importados

Además, dentro del módulo de Scipy encontramos diferentes funciones que nos han sido de utilidad:

Scipy	Estadísticos	Transformada de Fourier	Tratamiento señales
	pearsonr	fft	hilbert
	skew	fftfreq	chirp
	kurtosis	ifft	find peaks
		ifftshift	

Funciones dentro del módulo de Scipy

2.5. Herramientas y funciones utilizadas en Matlab

1. Variational Mode Decomposition.
2. Fast Kurtogram.
3. Envelope spectrum.

Estas funciones pueden encontrarse en [5]-[7]. Además, se han probado diferentes Toolbox, entre las que podemos destacar:

1. Statistics and Machine Learning Toolbox.
2. Signal Processing Toolbox.
3. Wavelet Toolbox.
4. Fast Kurtogram.
5. DSP System Toolbox.
6. Minimum Entropy Deconvolution Multipack.

El principal interés en el uso de Matlab radica en la multitud de funciones ya implementadas con ejemplos explicativos, lo que nos ha permitido entender mejor cómo funcionan algunas herramientas útiles para el análisis de nuestros conjuntos de datos y poder utilizarlas en Python.

3. Análisis en el dominio de tiempo

3.1. Importación de datos y almacenaje

En la base de datos que hemos encontrado existen diferentes mediciones de los rodamientos con diferentes variaciones de velocidad. Nosotros nos centraremos en analizar aquellos conjuntos de datos en los que la velocidad aumenta. Esta variación de velocidad se ha realizado en un rodamiento sano (H), un rodamiento con fallo de carrera interna (I) y un rodamiento con fallo de carrera externa (O). Para cada rodamiento se han realizado tres mediciones. Así, los conjuntos se denotarán mediante: 'H/I/O', dependiendo del rodamiento, 'A', puesto que se produce aceleración y '1/2/3' dependiendo de la medición. Una vez importados los archivos de datos en Jupyter, nos ayudaremos de la librería Pandas para la visualización de los datos (ver [4]).

Pandas es una librería utilizada para el análisis de datos que cuenta con las estructuras de datos que necesitamos para limpiar los datos en bruto para poder realizar su posterior análisis. Es una librería muy importante para procesar datos a alto nivel y fue diseñada originalmente para gestionar datos financieros, como alternativa al uso de hojas de cálculo. La estructura de datos básica de Pandas se denomina DataFrame: colección ordenada de columnas con nombres y tipos, similar a una tabla de base de datos. Una sola fila representa un único caso y las columnas representan atributos particulares. En este caso los elementos en distintas columnas pueden ser de diferentes tipos.

Así, en nuestro caso obtenemos:

```
t = np.linspace(0,10,2000000)
d = {'Tiempo': t,
     'Canal 1 H-1': canal1_11, 'Canal 1 I-1': canal1_21, 'Canal 1 O-1': canal1_31,
     'Canal 1 H-2': canal2_11, 'Canal 1 I-2': canal2_21, 'Canal 1 O-2': canal2_31,
     'Canal 1 H-3': canal3_11, 'Canal 1 I-3': canal3_21, 'Canal 1 O-3': canal3_31,

     'Canal 2 H-1': canal1_12, 'Canal 2 I-1': canal1_22, 'Canal 2 O-1': canal1_32,
     'Canal 2 H-2': canal2_12, 'Canal 2 I-2': canal2_22, 'Canal 2 O-2': canal2_32,
     'Canal 2 H-3': canal3_12, 'Canal 2 I-3': canal3_22, 'Canal 2 O-3': canal3_32}
canales=pd.DataFrame(d)
canales.head()
```

	Tiempo	Canal 1 H-1	Canal 1 I-1	Canal 1 O-1	Canal 1 H-2	Canal 1 I-2	Canal 1 O-2	Canal 1 H-3
0	0.000000e+00	-2.199445e-03	2.487109e-01	-7.460998e-03	-2.528292e-03	-1.699756e-02	2.075568e-03	1.746721e-03
1	5.000003e-06	-1.870597e-03	2.490398e-01	-6.474457e-03	-2.199445e-03	-1.436679e-02	2.733262e-03	7.601794e-04
2	1.000001e-05	-1.541750e-03	2.483821e-01	-6.474457e-03	-1.870597e-03	-1.140716e-02	4.313323e-04	-4.501374e-03
3	1.500001e-05	-1.870597e-03	2.503552e-01	-5.816763e-03	-1.212903e-03	-9.434081e-03	7.601794e-04	1.024852e-04
4	2.000001e-05	-2.199445e-03	2.510129e-01	-5.159068e-03	-3.185986e-03	-7.789845e-03	4.313323e-04	2.404415e-03

DataFrame

3.2. Correlación

Al encontrarnos con la medición de dos canales en cada conjunto de datos nos preguntamos si existiría alguna clase de relación por lo que decidimos calcular el coeficiente de correlación de Pearson obteniendo los siguientes resultados:

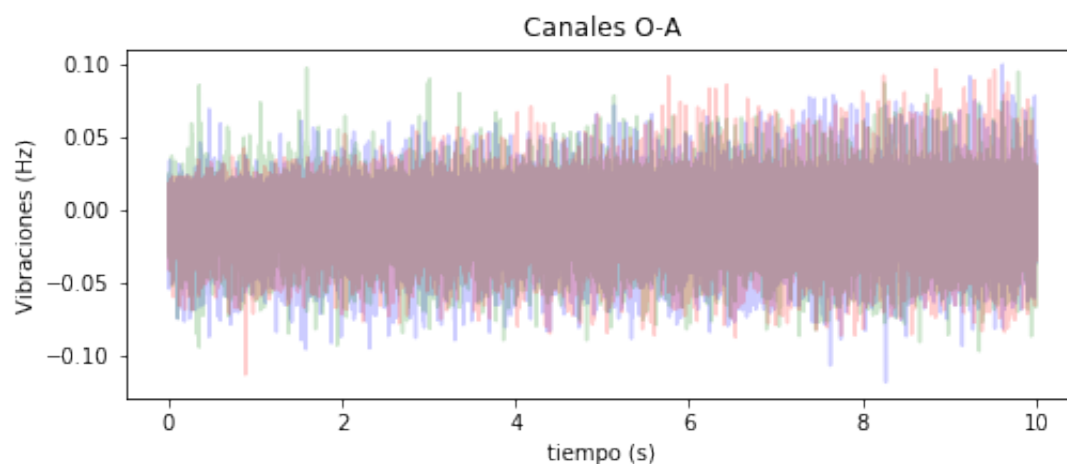
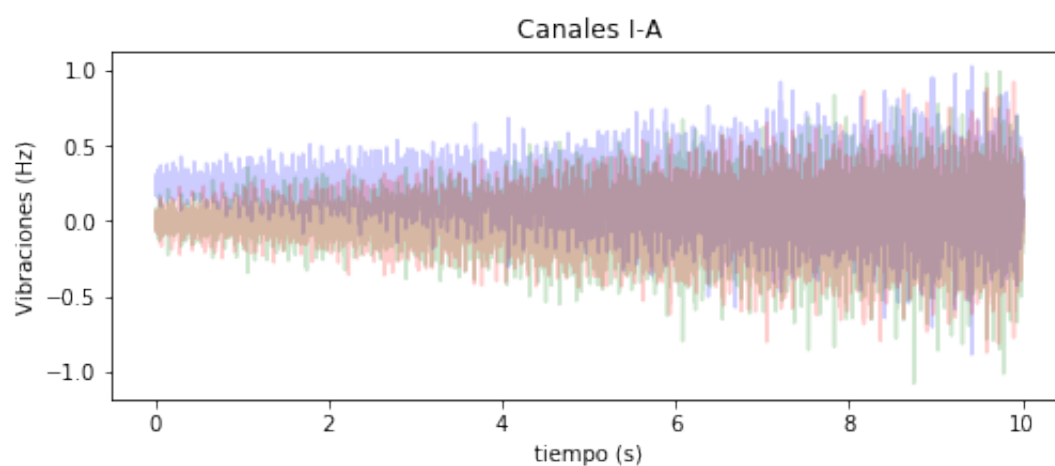
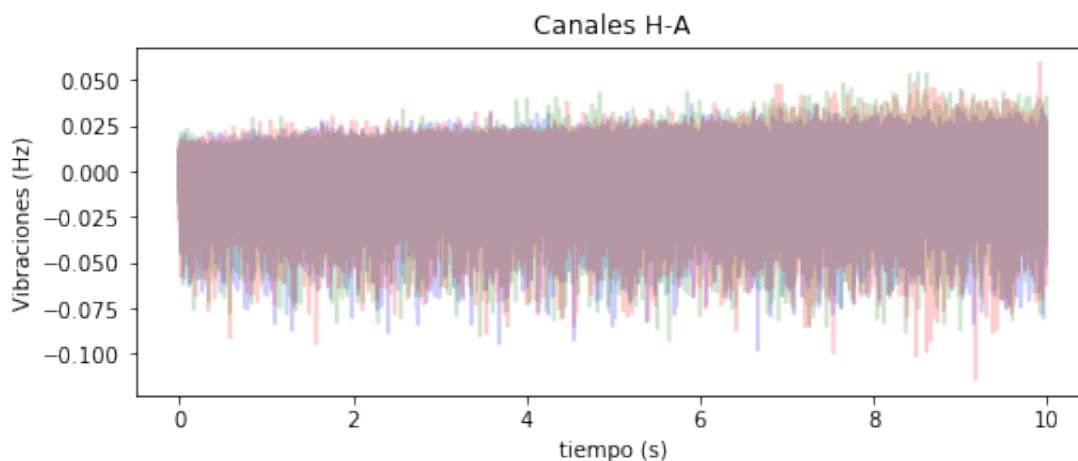
Coeficiente Pearson	H1	H2	H3	I1	I2	I3	O1	O2	O3
r	4.167338e-04	-9.405664e-06	-4.473436e-04	2.497785e-03	-4.055448e-04	-6.724663e-05	-3.308228e-04	-2.919106e-04	4.083706e-04

Coeficiente de correlación de Pearson

Donde r es el valor del índice de correlación, que varía entre $[-1, 1]$. Si el valor es próximo a 1 existe una correlación positiva (cuando una de las variables aumenta la otra también en proporción constante) y si es próximo a -1 existe correlación negativa (cuando una de las variables aumenta la otra disminuye). En nuestro caso es bastante próximo a cero por lo que puede considerarse que no existe relación lineal. Esto no necesariamente implica que los canales son independientes puesto que pueden existir relaciones no lineales entre ellos. En un principio procedemos a analizar sólo el Canal 1.

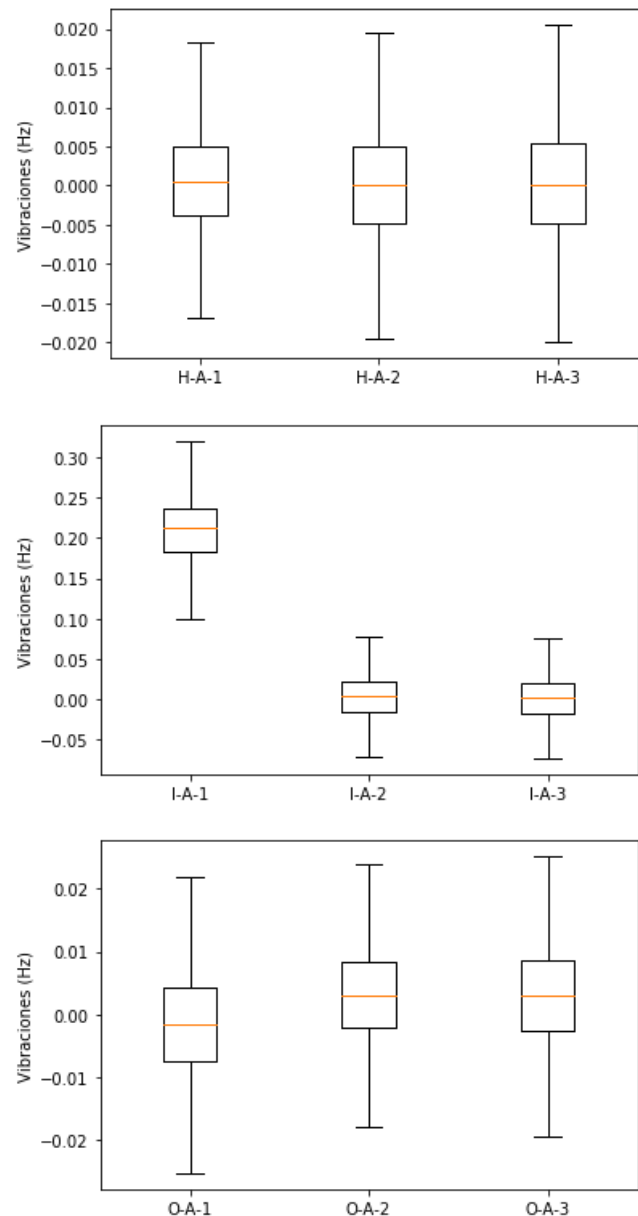
3.3. Representación

Se muestran a continuación en tres gráficas distintas las vibraciones con respecto del tiempo de los tres canales sanos, los tres con fallo de carrera interna y los tres con fallo de carrera externa con el fin de poder observar diferencias entre los canales.



A simple vista es posible observar diferencias entre las tres gráficas, especialmente en la que representa las vibraciones de los rodamientos con fallo de carrera interna.

Además de esta representación, se han realizado boxplots de las vibraciones de cada canal para poder visualizar la distribución de las señales de la forma más simplificada posible.



Representación gráfica de datos a través de sus cuartiles utilizando boxplots

Es posible observar claras diferencias entre aquellos canales sanos y con fallo de carrera interna. Sin embargo, no existen grandes evidencias que nos ayuden a concluir y diferenciar entre un canal sano y uno con fallo de carrera externa.

3.4. Relación entre los canales: índice de Jaccard

Sean A y B dos conjuntos, se define el índice de Jaccard entre A y B como:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (4)$$

El índice de Jaccard mide el grado de similitud entre dos conjuntos.

Es importante guardar cada canal como un conjunto. Después basta con realizar las siguientes operaciones:

```
#Indice de Jaccard canales HA1 y HA2

#1: Interseccion
inth12 = scanalh1_1 & scanalh1_2
#2: Union
unh12 = scanalh1_1 | scanalh1_2

#INDICE:
ijach12 = len(inth12)/len(unh12)
```

Cálculo del índice de Jaccard

A continuación mostramos dos tablas: en la primera tabla mostramos el índice de Jaccard entre los distintos canales. Recordemos que tenemos tres tipos de rodamientos: sano (H), con fallo de carrera interna (I) y con fallo de carrera externa (O) y que para cada canal se han realizado 3 distintas mediciones por lo que mostraremos el grado de similitud entre 9 conjuntos de datos. En la segunda tabla mostramos el grado de similitud que existe en media entre los canales sanos y con fallo de carrera interna, sanos y con fallo de carrera externa y por último entre aquellos que presentan fallo de carrera interna con fallo de carrera externa.

Conjuntos		HA1	HA2	HA3	IA1	IA2	IA3	OA1	OA2	OA3
0	HA1	1.000000	0.807512	0.834135	0.111805	0.104404	0.107025	0.665406	0.672481	0.628623
1	HA2	0.807512	1.000000	0.869565	0.104404	0.120152	0.123169	0.757519	0.756705	0.723327
2	HA3	0.834135	0.869565	1.000000	0.107025	0.118110	0.121076	0.734579	0.723327	0.695341
3	IA1	0.111805	0.128670	0.126483	1.000000	0.685562	0.683466	0.163335	0.157714	0.168957
4	IA2	0.104404	0.120152	0.118110	0.685562	1.000000	0.787807	0.152523	0.147273	0.157772
5	IA3	0.107025	0.123169	0.121076	0.683466	0.787807	1.000000	0.156353	0.150972	0.161734
6	OA1	0.665406	0.757519	0.734579	0.163335	0.152523	0.156353	1.000000	0.879342	0.886525
7	OA2	0.672481	0.756705	0.746641	0.157714	0.147273	0.150972	0.879342	1.000000	0.877917
8	OA3	0.628623	0.723327	0.695341	0.168957	0.157772	0.161734	0.886525	0.877917	1.000000

Índice de Jaccard entre todos los canales

Rodamientos		H	I	O
0	H	0.837071	0.117877	0.708958
1	I	0.117877	0.718945	0.157404
2	O	0.708958	0.157404	0.881261

Índice de Jaccard entre los distintos tipos de rodamientos

Es posible observar que entre los rodamientos sanos y aquellos con fallo de carrera interna existe una similitud menor del 20 % y entre los rodamientos sanos y los rodamientos con fallo de carrera externa una similitud menor del 80 %. Es importante destacar, además, que entre los propios canales con fallo de carrera interna no existe una similitud mayor del 80 %.

3.5. Estadísticos principales

Se han analizado varios estadísticos en los diferentes canales con el fin de encontrar diferencias. Estos han sido:

1. Varianza.
2. Momentos centrados de órdenes 3 y 4: en estadística el momento central o centrado de orden k de una variable aleatoria X se define como $E[(X - E[X])^k]$ donde E es el operador de la esperanza.
3. Skewness: medida de la asimetría de los datos. Dentro del módulo de estadísticos de Python, se calcula de la siguiente forma:

$$Skew = \frac{3(media - mediana)}{\sigma} \quad (5)$$

donde σ es la desviación estándar.

4. Kurtosis: la kurtosis de una variable estadística/aleatoria es una característica de forma de su distribución de frecuencias/probabilidad. Si esta es grande implica mayor concentración de valores de la variable cerca de la media de la distribución (pico) y muy lejos de ella (colas), al tiempo que existe menor frecuencia de valores intermedios. Una mayor kurtosis no implica una mayor varianza, ni al revés. Se calcula de la siguiente forma:

$$\beta_2 = \frac{\mu_4}{\sigma^4} \quad (6)$$

donde μ_4 denota el momento centrado de orden 4 y σ la desviación estándar.

Mostramos a continuación los estadísticos medidos en dos de los canales, uno sano y otro con fallo de carrera externa, en busca de diferencias:

	Estadísticos	H1	O1
0	Varianza	3.794782e+02	1.714595e+02
1	Momento de orden 3	2.009623e+02	9.686570e+01
2	Momento de orden 4	3.152279e+14	3.888898e+14
3	Skewness	4.183790e-02	7.577792e-02
4	Kurtosis	8.873413e+05	7.702427e+05

Estadísticos en el dominio de tiempo

4. Análisis en el dominio de la frecuencia

El análisis espectral de una señal tiene como objetivo la descomposición de la señal en sus componentes dentro del dominio frecuencial. Este análisis es una técnica ampliamente utilizada en varias especialidades de ingeniería, ciencias aplicadas y procesamiento de datos. Una tarea muy común en el análisis espectral es tratar de encontrar una determinada señal que está contaminada por otras, como por ejemplo ruido. Es posible que para ello necesitemos realizar algunos pasos antes de poder encontrar sus armónicos y poder decir qué clase de fallo contiene cada conjunto de datos.

4.1. Transformada rápida de Fourier

La transformada rápida de Fourier (FFT por sus siglas en inglés) es un algoritmo para evaluar de forma rápida la transformada de Fourier discreta y su inversa. Es muy importante en el análisis de señales por su rapidez de cálculo. Cualquier algoritmo para calcular la FFT contiene un conjunto de coeficientes espectrales, armónicos, que se pueden entender como muestras de la correspondiente función espectral continua (Transformada continua de Fourier) (ver [8]-[9]).

Es importante destacar dos partes de la FFT: amplitud y fase. Teniendo en cuenta que la FFT nos devuelve números complejos, la amplitud y la fase se calculan de la siguiente forma:

$$\text{Amplitud} = \sqrt{(\text{Re}(z))^2 + (\text{Im}(z))^2} \quad (7)$$

$$\text{Fase} = \text{atan}\left(\frac{\text{Im}(z)}{\text{Re}(z)}\right) \quad (8)$$

donde $z = a + bi \in \mathbb{C}$

Mostramos a continuación la implementación para el cálculo de la FFT:

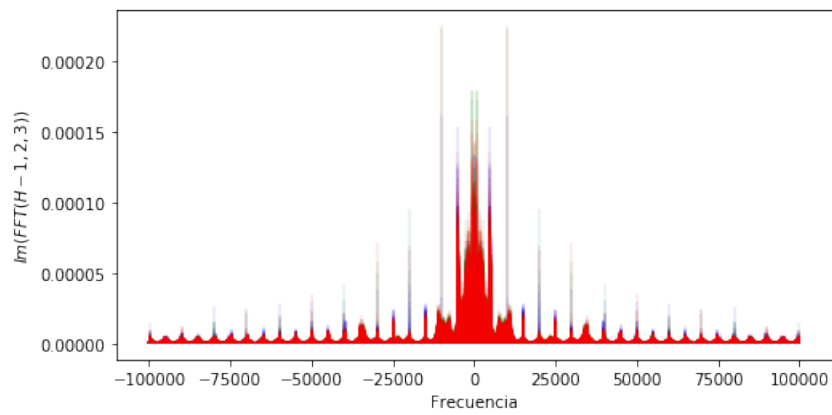
```
#Cálculo de cada FFT de cada una
from scipy.fft import fft, fftfreq, fftshift
#frecuencia
f = 200000

#Intervalo de muestreo
dt = t[1]-t[0]
n = len(canal01_1)

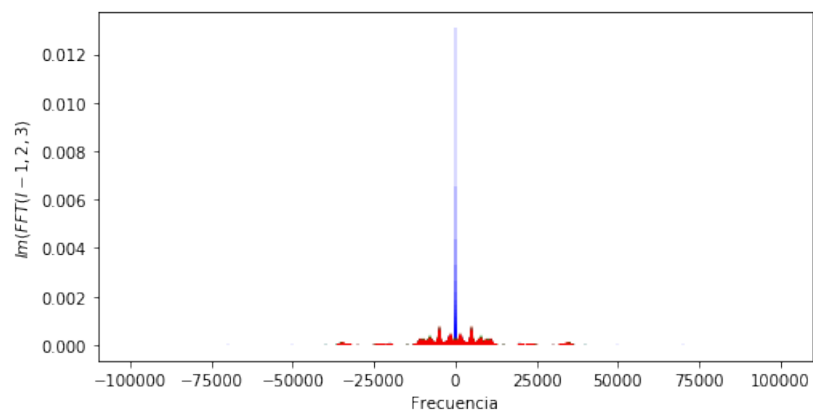
Xh_1 = fft(canalh1_1)/n
Xh_1 = np.fft.fftshift(Xh_1)
Xh_2 = fft(canalh1_2)/n
Xh_2 = np.fft.fftshift(Xh_2)
Xh_3 = fft(canalh1_3)/n
Xh_3 = np.fft.fftshift(Xh_3)
frec = fftfreq(n,dt)
frec = np.fft.fftshift(frec)
```

Cálculo de FFT

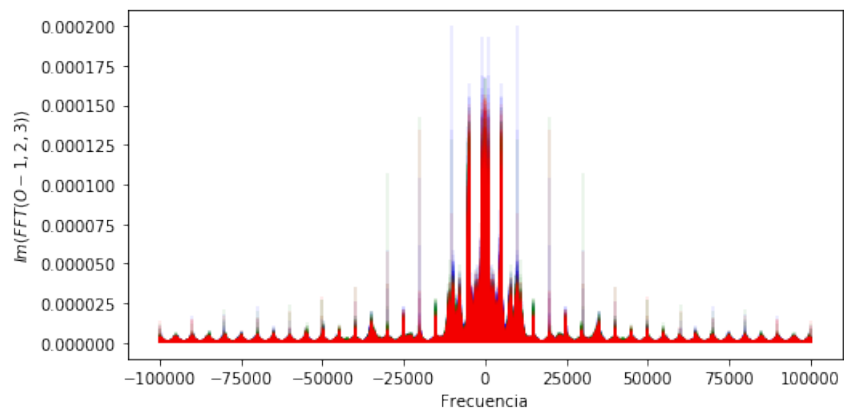
Representación:



FFT rodamiento sano



FFT Rodamiento con fallo carrera interna



FFT Rodamiento con fallo carrera externa

4.2. Estadísticos en el espectro

1. Centroide espectral: el centroide espectral es una medida utilizada en el procesamiento de señales digitales para caracterizar su espectro. Indica dónde se encuentra el “centro de gravedad” de éste. Su cálculo se realiza de la siguiente forma:

$$Centroid = \frac{\sum_0^{N-1} x(n)f(n)}{\sum_0^{N-1} x(n)} \quad (9)$$

donde $x(n)$ representa la magnitud del espectro y $f(n)$ la frecuencia central en cada punto.

2. Slope (pendiente espectral): en el procesamiento digital de señales es una medida de la rapidez con la que el espectro de un sonido de audio desciende hacia las frecuencias altas. Una forma de cuantificar esto es aplicando regresión lineal a la magnitud de la transformada rápida de Fourier de la señal, la cual produce un solo número que indica la pendiente de la línea que mejor se ajusta a través de los datos espectrales.

Mostramos a continuación los estadísticos realizados a dos canales, uno sano y otro con fallo de carrera externa en busca de posibles diferencias.

5	Centroid	5.000002e-01	5.000002e-01
6	Slope	7.452835e-08	<u>5.649048e-08</u>

Estadísticos en el dominio de tiempo

4.3. Filtrado de señales para eliminar fluctuaciones

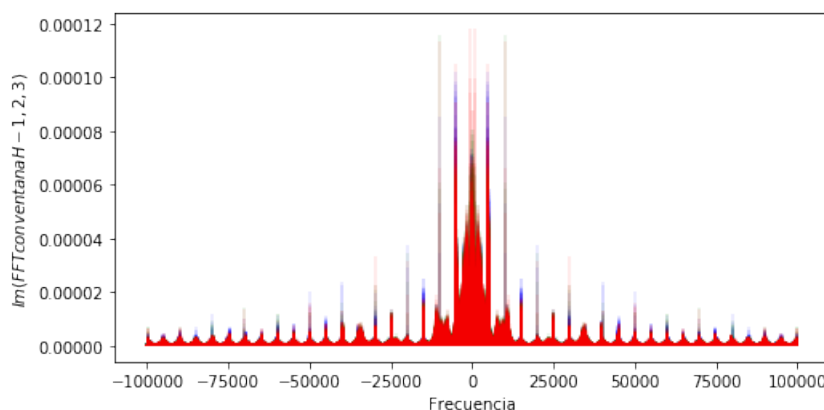
4.3.1. Ventanas

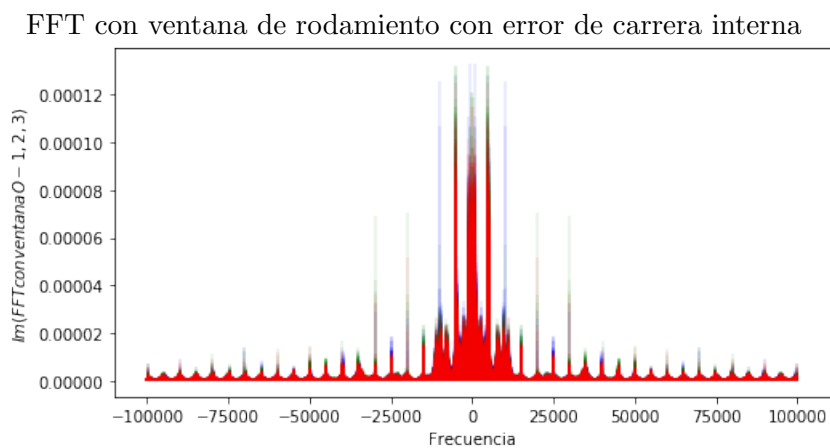
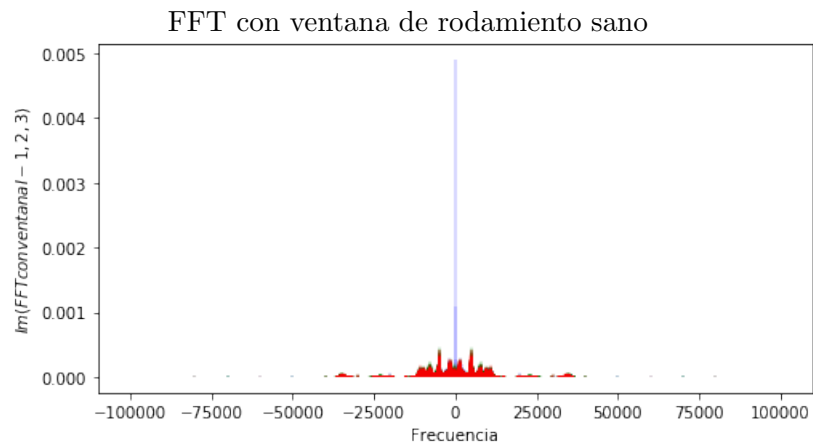
Las ventanas son funciones matemáticas usadas con frecuencia en el análisis y el procesamiento de señales para evitar las discontinuidades al principio y al final de los bloques analizados. Nosotros utilizamos la ventana Hanning. Esta ventana es satisfactoria en el 95 % de los casos. Tiene buena resolución de frecuencia y menor fuga espectral.

```
#Funciones ventana
xvh1=fftshift(fft(canalh1_1 * np.hanning(n))/n)
```

Ventana Hanning

Representación:





FFT con ventana de rodamiento con error de carrera externa

4.3.2. Cepstrum

El Cepstrum de una señal es el resultado de calcular la transformada de Fourier inversa del espectro de la señal estudiada en escala logarítmica. El cepstrum es complejo y, por tanto, tiene su parte real y su parte imaginaria. Es una herramienta útil para poder filtrar la señal en busca de sus armónicos eliminando posibles fluctuaciones de la señal. Mostramos a continuación el diagrama para su implementación (ver [15]):

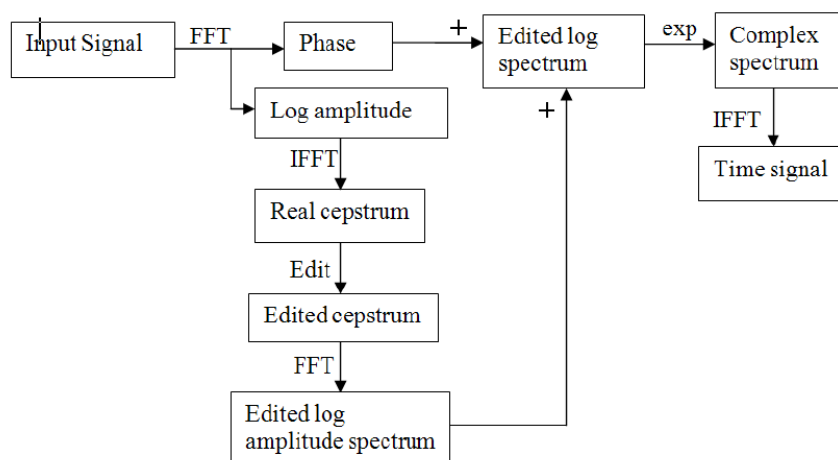


Diagrama Cepstrum

A continuación, los pasos que han sido realizados para su cálculo:

```
def filtradoCepstrum(input_signal, t, fs):
    dt = t[1]-t[0]
    n = len(input_signal)
    fftsignal = fft(input_signal)
    frec = fftfreq(n,dt)

    #Calculamos la fase y la amplitud
    fase = np.unwrap(np.angle(fftsignal))
    amplitud = abs(fftsignal)

    #Manipulamos la amplitud:
    logampli = np.log(amplitud)
    #Obtenemos el ceptrum real:
    realceps = np.fft.ifftshift(logampli)

    #Realizamos el filtro de paso bajo
    edit_rcep = highlowfilter('low', realceps)
    #Realizamos la FFT y nos quedamos con la amplitud
    editampli = abs(np.fft.fftshift(edit_rcep))
    edit_ampli = np.log(editampli)

    #Unimos la fase y la amplitud
    unionedit = fase*1j + edit_ampli

    #Realizamos la exponencial de este número para obtener el complejo en forma exponencial
    cepscm = []
    for i in range(0,len(unionedit)):
        cepscm.append(cmath.exp(unionedit[i]))
    cepscm = np.array(cepscm)

    #Realizamos la funcion inversa y obtenemos la señal filtrada
    filtradatotal = np.fft.ifft(cepscm)
    filtradareal = filtradatotal.real

    return filtradatotal, filtradareal
```

Pasos para la implementación de Cepstrum

4.3.3. Otros filtros: paso alto/bajo

Es posible que el ruido aleje las frecuencias de la señal que tenemos de las frecuencias que deseamos por lo que puede ser conveniente realizar un filtro para quedarnos con aquellas frecuencias que nos interesan. Los filtros de paso bajo atenúan las frecuencias altas y dejan pasar las frecuencias bajas y los filtros de paso alto realizan justo lo contrario. Estos filtros pueden ser útiles a la hora de determinar si el fallo es de carrera interna (filtro de paso bajo) o de carrera externa (filtro de paso alto) (ver [14]).

En nuestro caso utilizamos lo siguiente:

```
from scipy import signal
import matplotlib.pyplot as plt

def highlowfilter(type, input_signal):
    #type: 'hp' high pass, 'low': low pass

    b, a = signal.butter(3, 0.05, type)

    zi = signal.lfilter_zi(b, a)
    z, _ = signal.lfilter(b, a, input_signal, zi=zi*input_signal[0])

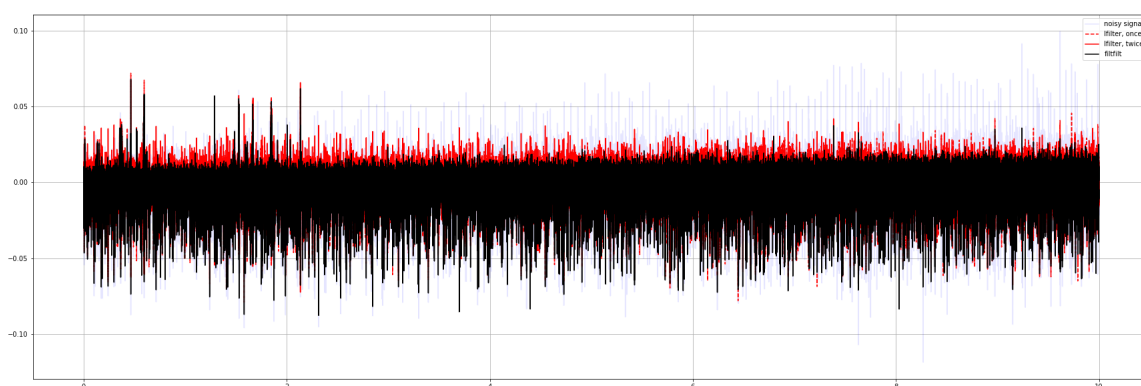
    z2, _ = signal.lfilter(b, a, z, zi=zi*z[0])

    y = signal.filtfilt(b, a, input_signal)

    fig = plt.figure(figsize=(30,10))
    plt.plot(t, canal1_31, 'b', alpha=0.1)
    plt.plot(t, z, 'r--', t, z2, 'r', t, y, 'k')
    plt.legend(('noisy signal', 'lfilter, once', 'lfilter, twice',
               'filtfilt'), loc='best')
    plt.grid(True)
    plt.show()
```

Pasos para la implementación del filtro de paso alto/bajo

Mostramos a continuación una imagen de un filtro de paso alto en uno de los datasets que contiene fallo de carrera externa.

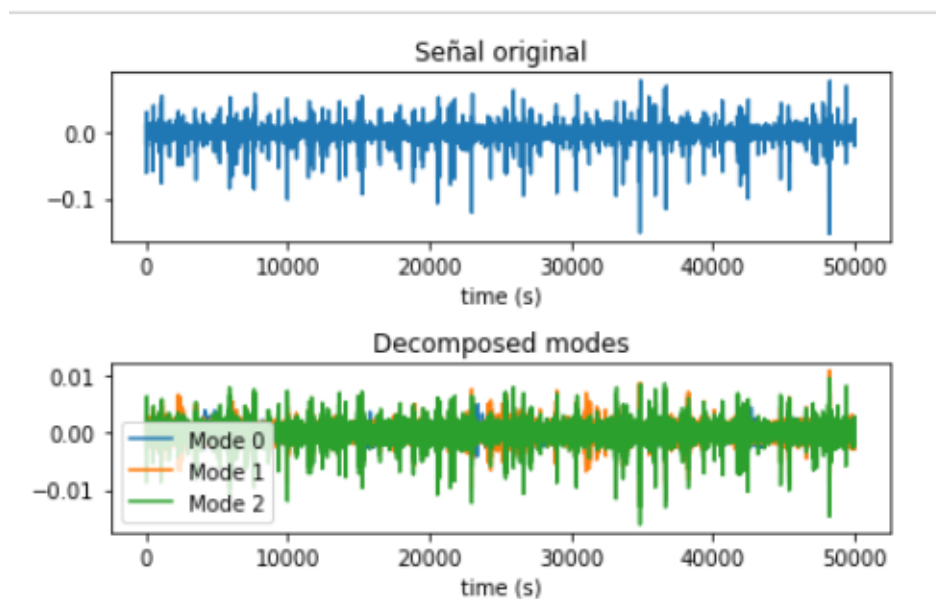


Filtro de paso alto canal O-A-1

En la imagen se muestra la señal original (en morado, traslúcido) y los diferentes filtros. El filtro final se muestra en negro.

4.4. Descomposición de la señal: Variational Mode Decomposition (VMD)

Existen varias herramientas para la descomposición de una señal: la Descomposición Empírica en Modos (Empirical Mode Decomposition, EMD) es un método guiado por los datos que permite expresar la señal como suma de componentes AM-FM. Cada una de estas componentes, llamadas funciones de modo intrínsecas (IMF), representa procesos físicos subyacentes en la señal. Gracias a este método podemos eliminar ruido de la señal y así poder centrarnos en aquella parte que nos interesa(ver [10]-[13]). En nuestro caso se decidió utilizar otro método, Variational Mode Decomposition, más actual y eficiente y que reemplaza la descomposición empírica. En este método los modos se extraen de forma no recursiva.

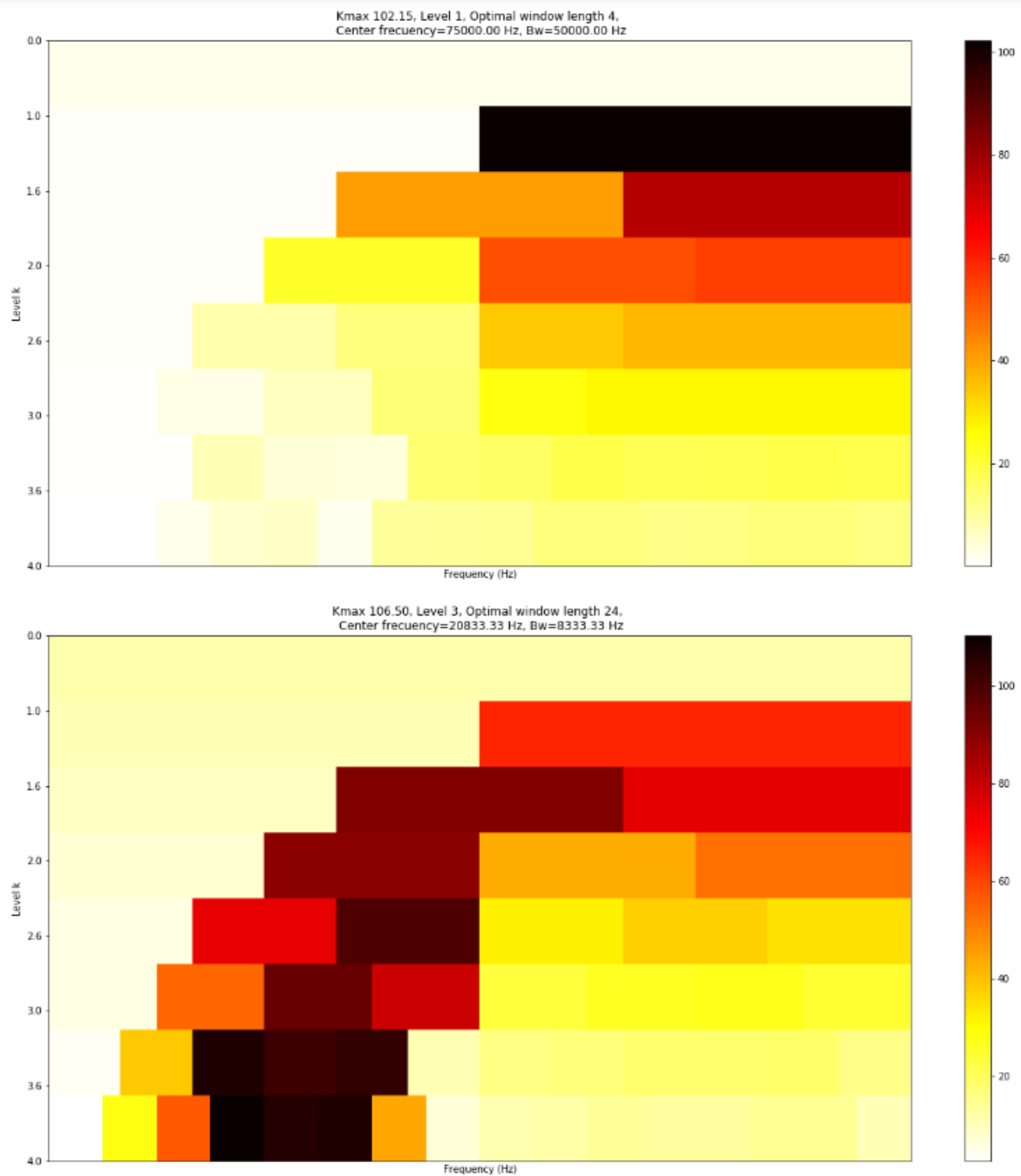


Descomposición señal en tres modos distintos

4.5. Kurtosis espectral y kurtograma

La kurtosis espectral (SK) se basa en el filtrado de paso de banda y se utiliza para filtrar la señal de vibración del rodamiento para detectar fallos en éste. Los métodos para el cálculo de SK aplican ventanas de longitud no uniforme en el dominio de la frecuencia. Es una herramienta especialmente útil cuando la velocidad es constante, pero puede adaptarse a casos de velocidad variable una vez la señal ha sido filtrada. Para mejorar la eficiencia, se utiliza el kurtograma rápido (fast-kurtogram)(ver [6]). Este combina un árbol binario 1/3 para descomponer la señal y posteriormente calcular SK.

El kurtograma nos permite obtener una banda óptima en el dominio de la frecuencia en la que buscar los armónicos para intentar detectar los posibles fallos, haciendo el análisis menos costoso. Para ello, se han importado varios métodos desde Matlab, que pueden encontrarse en [4]. Se muestra a continuación ejemplos de kurtograma.

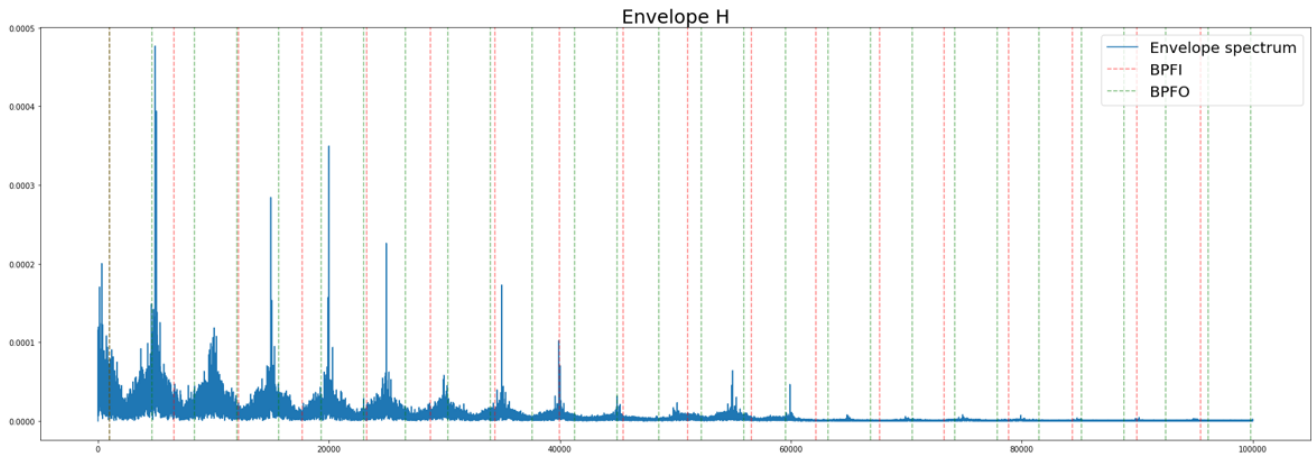


Kurtosis Canales H e I

En aquellos canales con fallo de carrera interna sólo debemos centrarnos en analizar aquellas frecuencias donde se alcanza la mayor kurtosis que, como podemos observar son distintas a un canal sano.

4.6. Envolvente espectral

Una envolvente espectral es una curva en el plano de amplitud de frecuencia, derivada de un espectro de magnitud de Fourier. El método utilizado está basado en varias funciones que pueden encontrarse en [5]. A continuación, mostramos un ejemplo del análisis de un canal sano utilizando esta herramienta:



Envelope canal sano

Una vez tenemos la imagen filtrada, realizamos la envolvente del espectro. Puede verse en la imagen representado en líneas verticales discontinuas los armónicos correspondientes a errores de carrera externa (BPFO, en verde) y de carrera interna (BPFI, en rojo). Como puede observarse en esta imagen, no existen grandes coincidencias con estos armónicos.

5. GitHub

GitHub es una plataforma creada para que los desarrolladores puedan subir el código de sus aplicaciones y herramientas. Como resultado de estas prácticas puede encontrarse en [1] distintas funciones y conjuntos de datos en los que se muestran, de forma resumida, los distintos procedimientos y funciones que se han realizado durante este período. Además, para facilitar la ejecución de las distintas funciones, pueden encontrarse también los conjuntos de datos que han sido utilizado para ejecutar las funciones. Todos los conjuntos de datos pueden encontrarse en [3].

1. PuestaenComun: en él se expone:

- a) Importación de datos.
- b) Muestra de los canales en un dataframe.
- c) Correlación de los dos canales.
- d) Representación de los canales.
- e) Boxplot de los canales.
- f) Cálculo de la FFT y representación.
- g) FFT con ventana y representación.
- h) Estadísticos principales de los canales.

2. PruebaJaccard: en éste se exponen los resultados de realizar el índice de Jaccard a los diferentes canales.

3. Funciones: Jupyter Notebook con las principales funciones implementadas a lo largo de este período. Al principio se encuentra un índice de las funciones más utilizadas y cómo llamar a cada una de ellas. Entre ellas destacamos:

- a) Estadísticos: Slope, Centroide.
- b) Tabla con los principales estadísticos de la señal deseada.
- c) Mensaje con kurtosis.
- d) Filtro de paso alto/bajo.
- e) Cepstrum de la señal.
- f) Kurtograma.
- g) Envelope spectrum y su gráfica. Porcentaje de comunes con los armónicos BPFI y BPFO.
- h) Descomposición de la señal utilizando VMD
- i) Gráfica de la FFT de la señal deseada y los armónicos BPFI y BPFO.

4. LlamadaFunciones: utilizamos las funciones que se encuentran en *Funciones*. En él se expone:
- a)* Importación de canales.
 - b)* Muestra de la kurtosis de cada canal.
 - c)* Fast kurtogram y kurtograma de cada canal.
 - d)* Cepstrum y representación del canal junto a su señal filtrada.
 - e)* Envelope spectrum de las señales filtradas.
 - f)* Porcentaje de armónicos comunes de cada señal (BPFI, BPFO).
 - g)* Descomposición en IMFs utilizando VMD para el canal sano (H).

6. Valoración final

Mi valoración final de estas prácticas es claramente positiva. Cabe destacar que a pesar de las circunstancias y habiendo realizado estas prácticas de forma telemática, he obtenido ayuda por parte de mis compañeros siempre que la he necesitado. Mi trabajo ha estado en constante supervisión durante todo este período, mediante informes y reuniones diarias y además he obtenido completa disponibilidad por parte de mi tutor y compañeros ante cualquier duda mediante Teams. Creo que entre las competencias que he desarrollado en este tiempo puedo destacar:

1. Autonomía: realizando el curso de aprendizaje de Python, buscando información sobre los distintos métodos para el tratamiento de señales, leyendo artículos e implementando distintas funciones en Jupyter y Matlab.
2. Trabajo en equipo: se han realizado reuniones diarias para que los distintos compañeros informemos de nuestro trabajo diario y así disminuir el distanciamiento causado por el teletrabajo. Esto ha permitido poder comentar los avances (y retrocesos) a la hora de analizar una señal y avanzar más rápido a la hora de trabajar, puesto que es más fácil ver errores, soluciones o indicar cómo seguir en la búsqueda de distintos métodos.
3. Búsqueda de información: como se ha citado anteriormente, se ha tenido que buscar información para saber qué métodos implementar y cómo. Varios de estos métodos pueden encontrarse en la bibliografía y además pueden encontrarse varios artículos cuya lectura ha sido de especial utilidad a la hora de analizar nuestro conjunto de datos.
4. Utilizar software científico: se han utilizado Matlab y Jupyter, como se ha citado anteriormente, para el análisis de las diferentes señales. Las múltiples herramientas disponibles en Matlab y su clara explicación han sido de gran utilidad para realizar el análisis en Python.
5. Reflexionar: saber aplicar los distintos métodos correspondientes y, a raíz de los resultados, comprender en qué casos funcionan y en cuáles no y porqué para poder continuar el análisis y buscar otros métodos más eficientes.
6. Desarrollar programas: se ha utilizado principalmente Jupyter, en el que se han importado funciones y otras se han creado para poder analizar los distintos datos.

Referencias

- [1] GitHub. Repositorio con funciones utilizadas.
<https://github.com/irenegz/signalprocessing/tree/master>
- [2] Coursera. *Python for everybody*, University of Michigan.
<https://www.coursera.org/specializations/python>
- [3] Huang, H., Baddour, N. Bearing vibration data collected under time-varying rotational speed conditions. *Data in Brief*, 21: 1745-1749. (Diciembre 2018)
<https://www.sciencedirect.com/science/article/pii/S2352340918314124>
- [4] Guía librería Pandas.
https://pandas.pydata.org/docs/user_guide/index.html
- [5] Mathworks. Variational Mode Decomposition.
<https://es.mathworks.com/help/wavelet/ref/vmd.html>
- [6] Mathworks. Fast Kurtogram.
<https://es.mathworks.com/matlabcentral/fileexchange/48912-fast-kurtogram>
- [7] Mathworks. Envelope Spectrum.
<https://www.mathworks.com/help/signal/ref/envspectrum.html>
- [8] Smith, J.O. III. *Mathematics of the discrete Fourier transform (DFT) with audio applications*, Center for Computer Research in Music and Acoustics (CCRMA)
<https://ccrma.stanford.edu/~jos/mdft/>
- [9] Wang, T., Liang, M., Li, J., Cheng, W. Rolling element bearing fault diagnosis via fault characteristic order (FCO) analysis. *Mechanical Systems and Signal Processing*. Volume 45, Issue 1: 139-153. (3 marzo 2014)
<https://www.sciencedirect.com/science/article/abs/pii/S0888327013006389>
- [10] Abdelkader, R., Kaddour, A., Derouiche, Z. Enhancement of rolling bearing fault diagnosis based on improvement of empirical mode decomposition denoising method. *Int J Adv Manuf Technol* **97**, 3099–3117 (2018). <https://doi.org/10.1007/s00170-018-2167-7>
- [11] Torres, M. E., Colominas, M. A., Schlotthauer, G., Flandrin, P. A complete ensemble empirical mode decomposition with adaptive noise. *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, pp. 4144-4147, (2011), doi: 10.1109/ICASSP.2011.5947265.
- [12] Du, W., Wang, Z., Gong, X., Wang, L., Luo, G., .°ptimum IMFs Selection Based Envelope Analysis of Bearing Fault Diagnosis in Plunger Pump”, *Shock and Vibration*, vol. 2016, Article ID 1248626, 8 page, (2016). <https://doi.org/10.1155/2016/1248626>
- [13] Guo, W., Tse, P.W., Djordjevich, A. Faulty bearing signal recovery from large noise using a hybrid method based on spectral kurtosis and ensemble empirical mode decomposition, *Measurement*, Volume 45, Issue 5, 1308-1322, (2012).
<https://doi.org/10.1016/j.measurement.2012.01.001>

-
- [14] Randall, R.B., Antoni, J., Rolling element bearing diagnostics-A tutorial. *Mechanical Systems and Signal Processing*, Volume 25, Issue 2, 485-520, (2011).
<https://doi.org/10.1016/j.ymssp.2010.07.017>
- [15] Randall, R.B., Sawalhi N. Use of the cepstrum to remove selected discrete frequency components from a time signal. In: Proulx T. (eds) *Rotating Machinery, Structural Health Monitoring, Shock and Vibration*, Volume 5. Conference Proceedings of the Society for Experimental Mechanics Series. Springer, New York, NY. (2011) https://doi.org/10.1007/978-1-4419-9428-8_38