



UNIVERSIDAD
DE GRANADA

Máster Universitario en Estructuras
Curso 2020-2021

Taller: Procesamiento de señales

Módulo: MÓDULO FUNDAMENTAL: CALIDAD Y DAÑO

Materia: Análisis Modal y Detección de Defectos

Enrique García Macías

enriquegm@ugr.es

**Departamento de Mecánica de Estructuras e
Ingeniería Hidráulica**

Desarrollo del curso

		FECHA		HORA	PROFESOR	TEMA	
Clase 1	Lunes	1	febrero	9:30-11:30	EGM	1	Introducción: Análisis modal dentro del marco del mantenimiento de la salud estructural.
Clase 2	Lunes	8	febrero	9:30-11:30	EGM	2	Fuentes de deterioro, patologías estructurales, y tecnologías de monitorización.
Clase 3	Lunes	15	febrero	9:30-11:30	EGM	3	Taller: procesamiento de señales.
Clase 4	Lunes	22	febrero	9:30-11:30	EGM	4	Análisis modal experimental.
Clase 5	Lunes	15	marzo	9:30-11:30	EGM	5	Análisis modal operacional.
Clase 6	Lunes	12	abril	9:30-11:30	EGM	6	Análisis modal operacional automatizado. Práctica de laboratorio I.
Clase 7	Lunes	19	abril	9:30-11:30	EGM	7	Taller: Identificación del daño estructural.
Clase 8	Lunes	26	abril	9:30-11:30	RCT	8	Técnicas de identificación dinámica basadas en análisis modal operacional.
Clase 9	Lunes	26	abril	12:00-14:00	RCT	9	Práctica de laboratorio II: Test de vibración ambiental.
Clase 10	Martes	27	abril	9:30-11:30	RCT	10	Casos de estudio.
Clase 11	Martes	27	abril	12:00-14:00	RCT		Presentación de trabajos.

ENTREGA DE TRABAJOS Y EVALUACIÓN

Del 3 al 28 de mayo

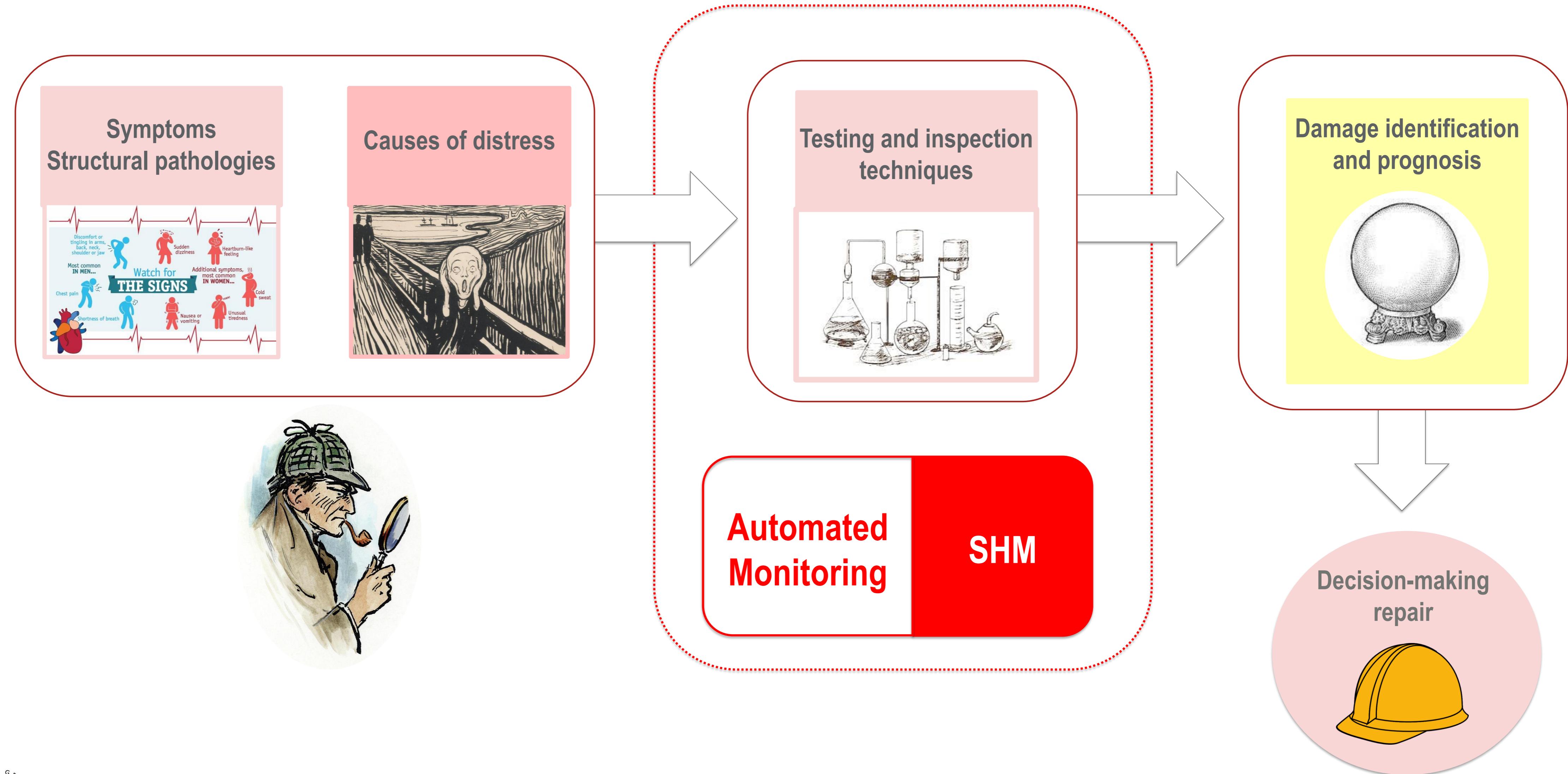


UNIVERSIDAD
DE GRANADA



Introducción.

Introducción



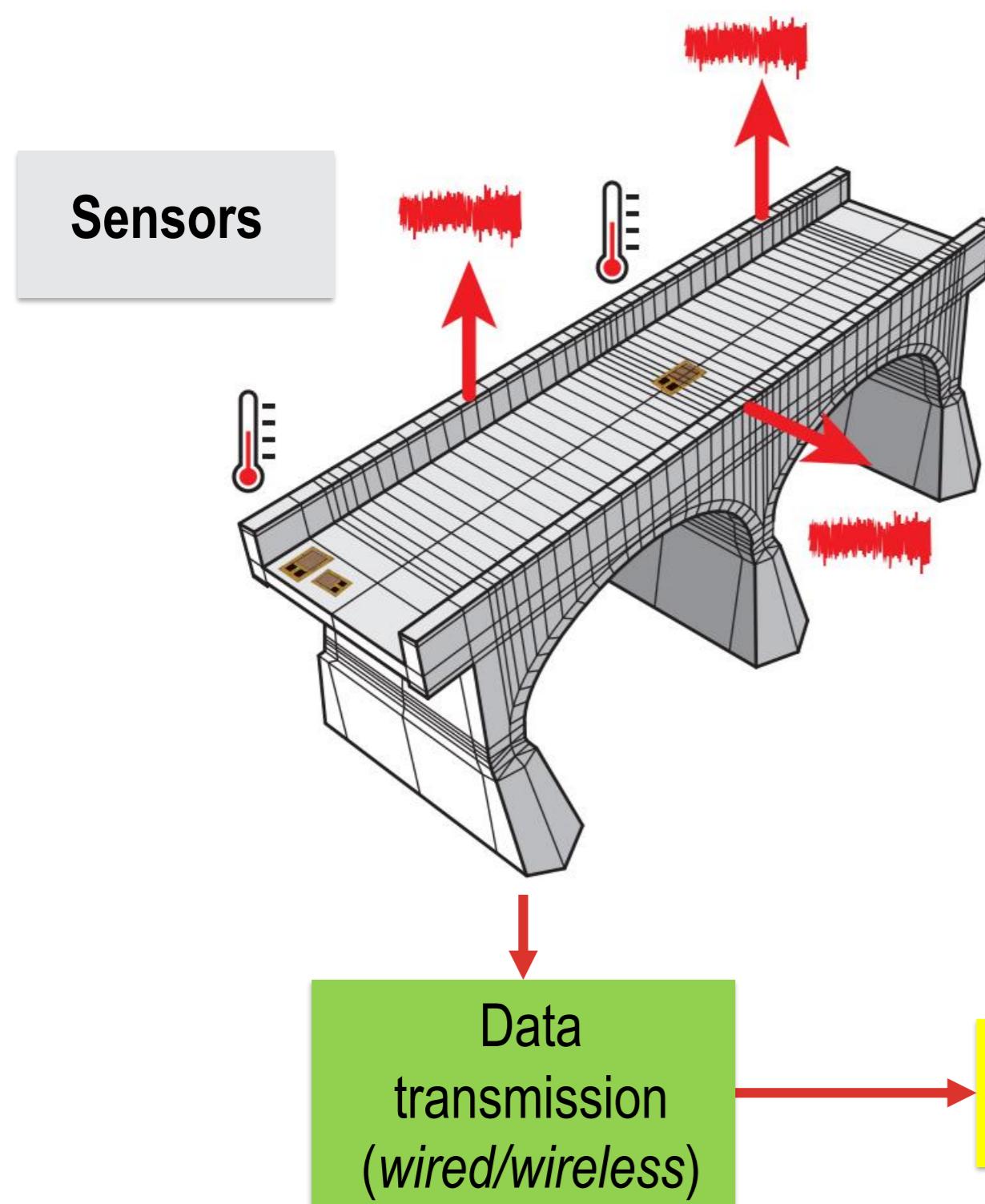
**Automated
Monitoring**

SHM

**Decision-making
repair**

Introducción

Monitored physical phenomenon



Signal Pre-processing

Feature extraction

Pattern recognition

Pattern classification

Prognosis

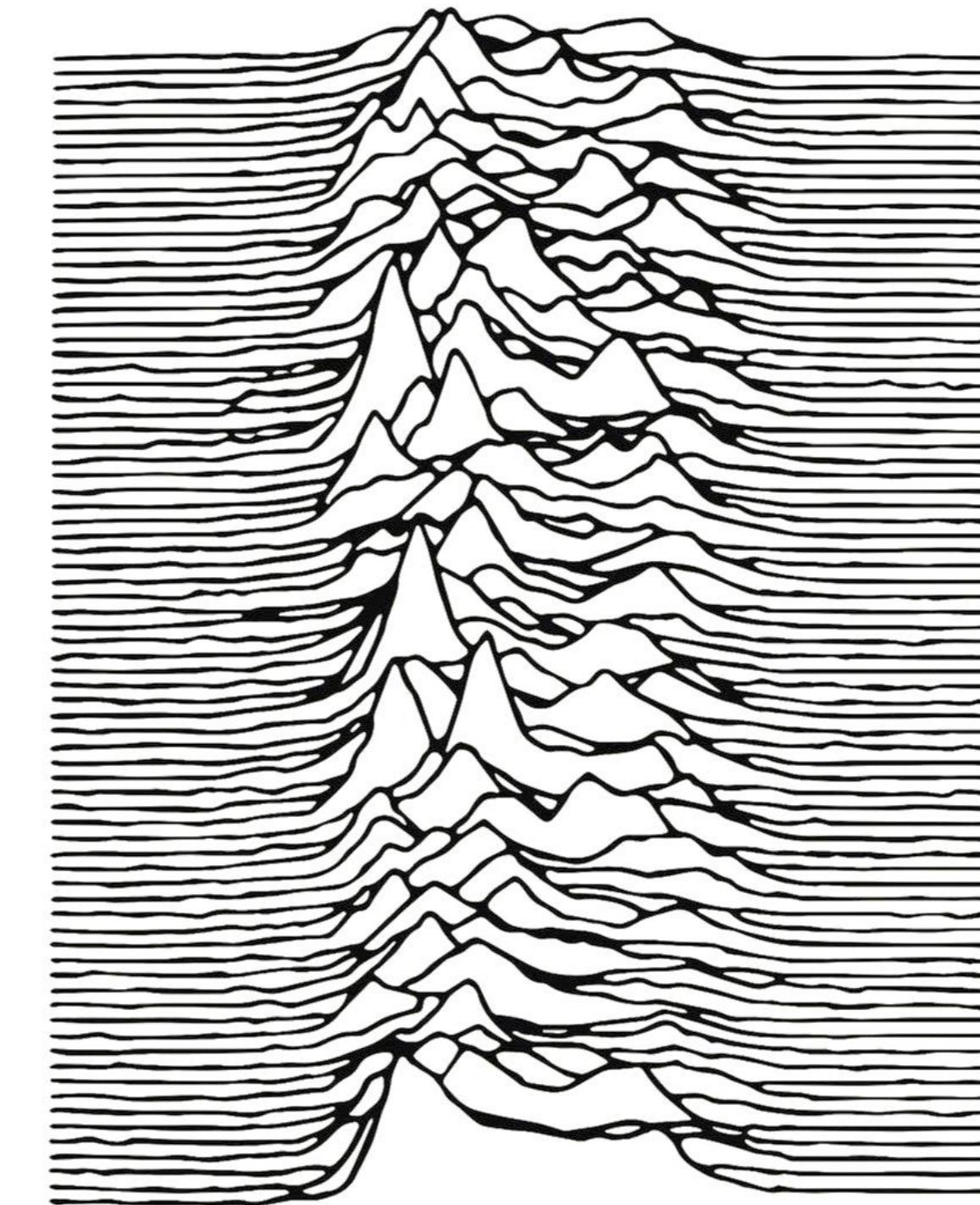
Decision

Damage detection

Damage identification

Outline

- ❖ Signals
- ❖ Reminder – Fourier Transform
- ❖ Aliasing
- ❖ Frequency leakage
- ❖ Cross-correlation
- ❖ Random signals
- ❖ Filtering
- ❖ Missing data
- ❖ Power-spectral density
- ❖ Time-frequency domain

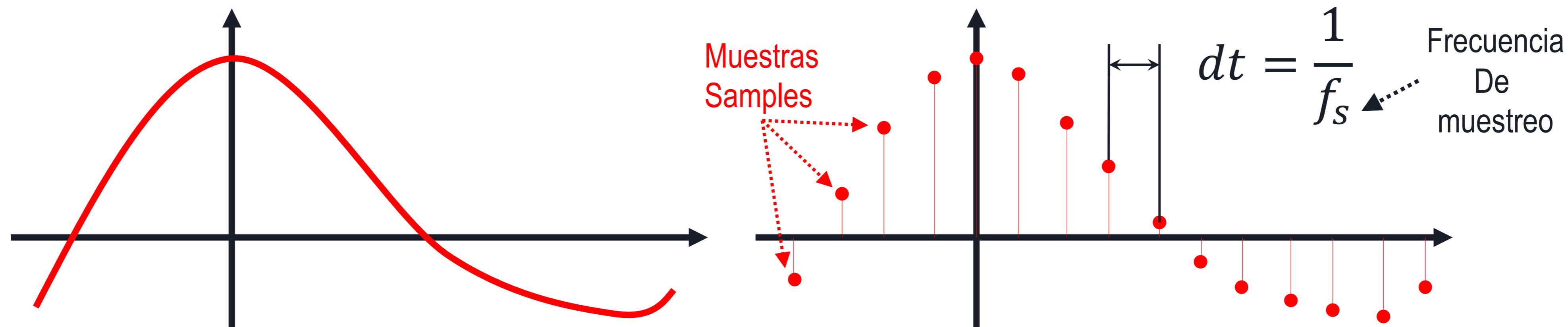




Señales.

Algunas definiciones básicas

- Una **señal** es una función que contiene información acerca del comportamiento de un sistema o los atributos de algún fenómeno.
- **El procesamiento de señales** es la actividad que opera sobre una señal de entrada para producir una señal de salida.
- Las señales se pueden clasificar en dos grupos: **señales continuas en el tiempo y señales discretas en el tiempo**.

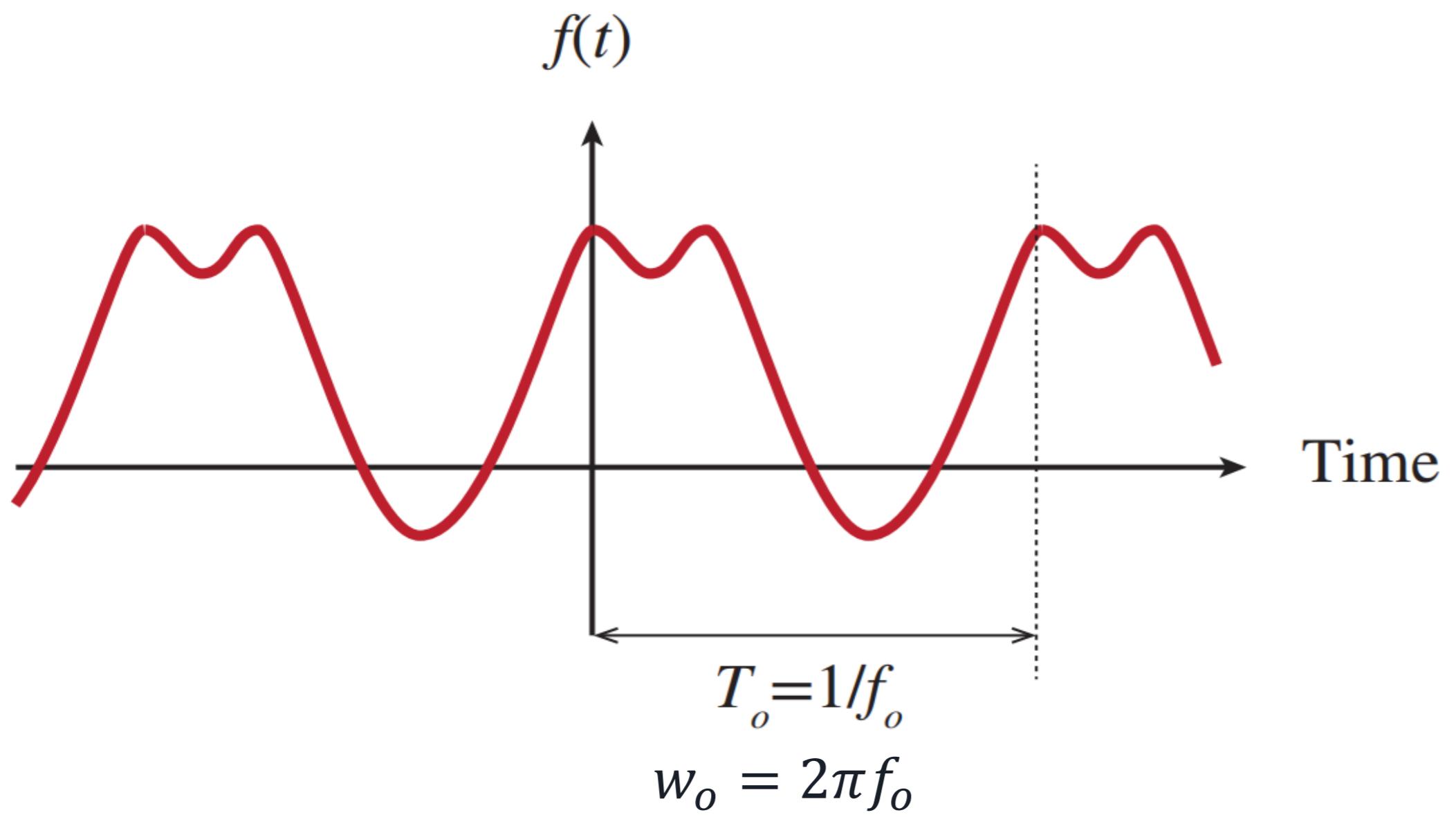




Señales – Recordatorio: Transformada de Fourier

Fourier Transform - Fundamentals

Una señal periódica $x(t)$ con período T_o es aquella que se repite cada T_o segundos. La serie de Fourier (Fourier series FS) es una descomposición de una señal periódica en una combinación lineal de funciones seno y coseno. Las frecuencias de estas funciones son múltiplos de la frecuencia fundamental de la señal $f_o = 1/T_o$.



$$\omega_n = 2\pi n f_o$$

$$x(t) = a_0 + \sum_{n=1}^{\infty} [a_n \cos(\omega_n t) + b_n \sin(\omega_n t)]$$

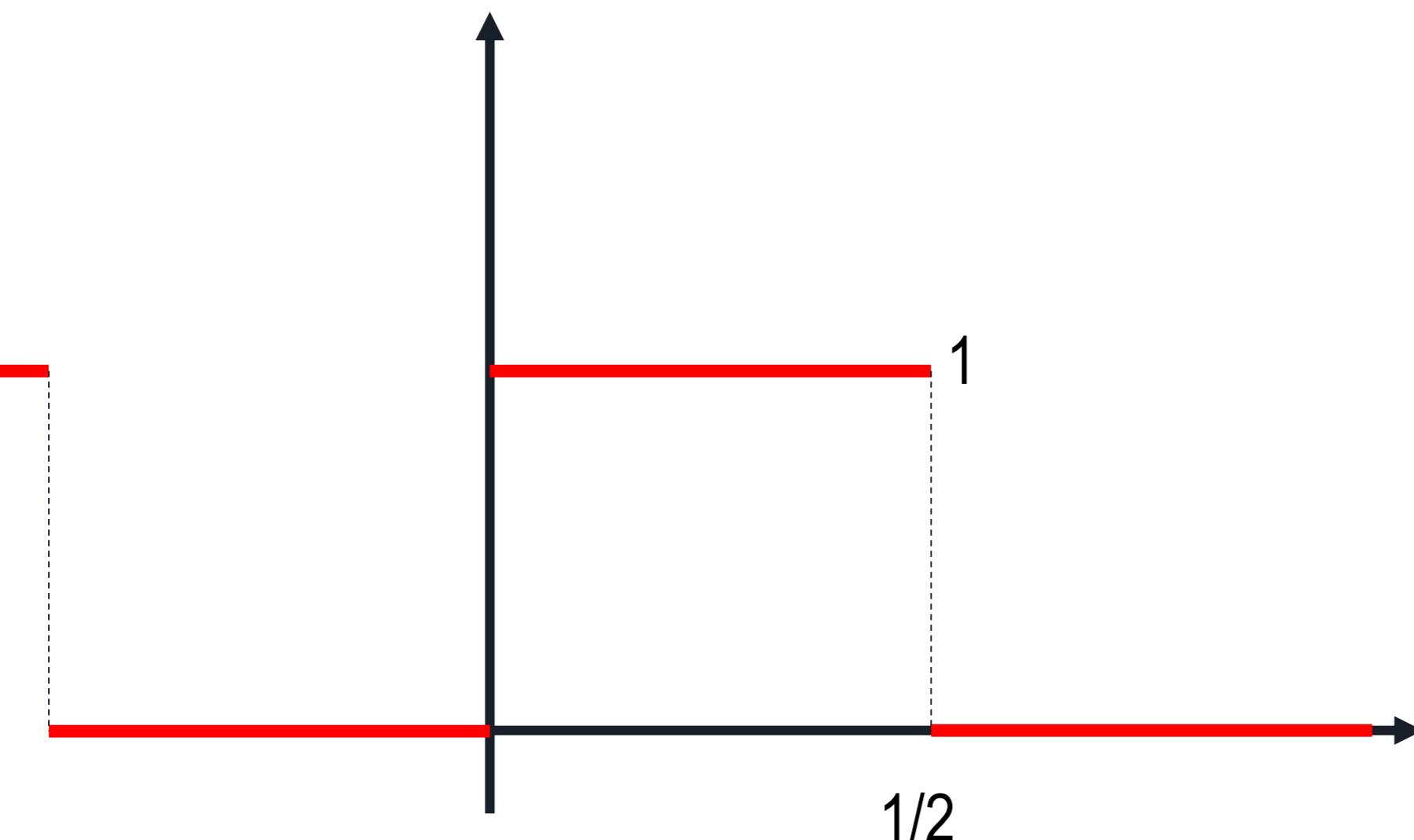
$$a_0 = \frac{1}{T_o} \int_0^{T_o} x(t) dt$$

$$a_n = \frac{2}{T_o} \int_0^{T_o} x(t) \cos(nw_o t) dt$$

$$b_n = \frac{2}{T_o} \int_0^{T_o} x(t) \sin(nw_o t) dt$$

Señales – Recordatorio: Transformada de Fourier

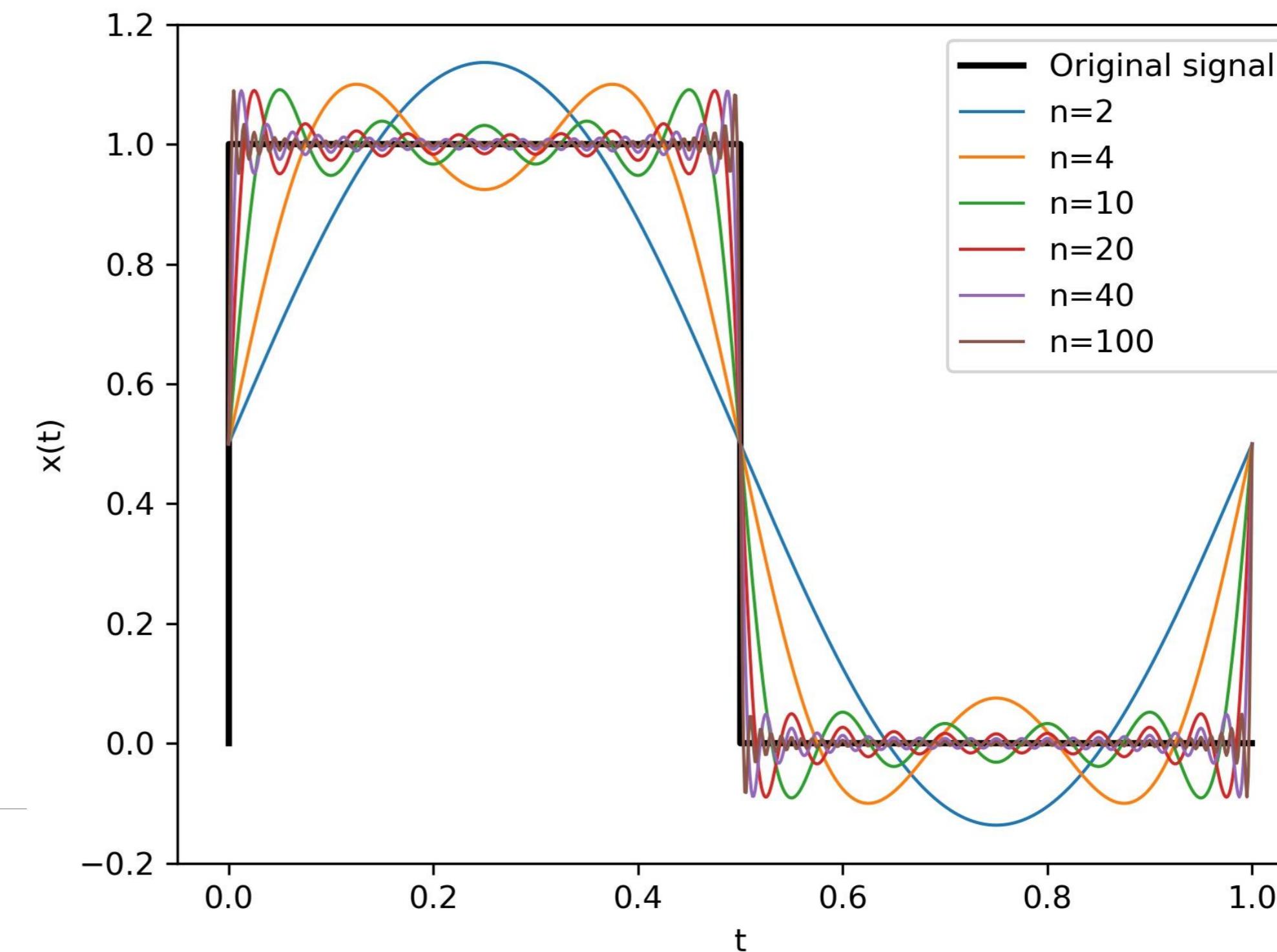
Ejemplo 1: señal de onda cuadrada



$$a_0 = \frac{1}{T_o} \int_0^{T_o} x(t) dt = 1/2$$

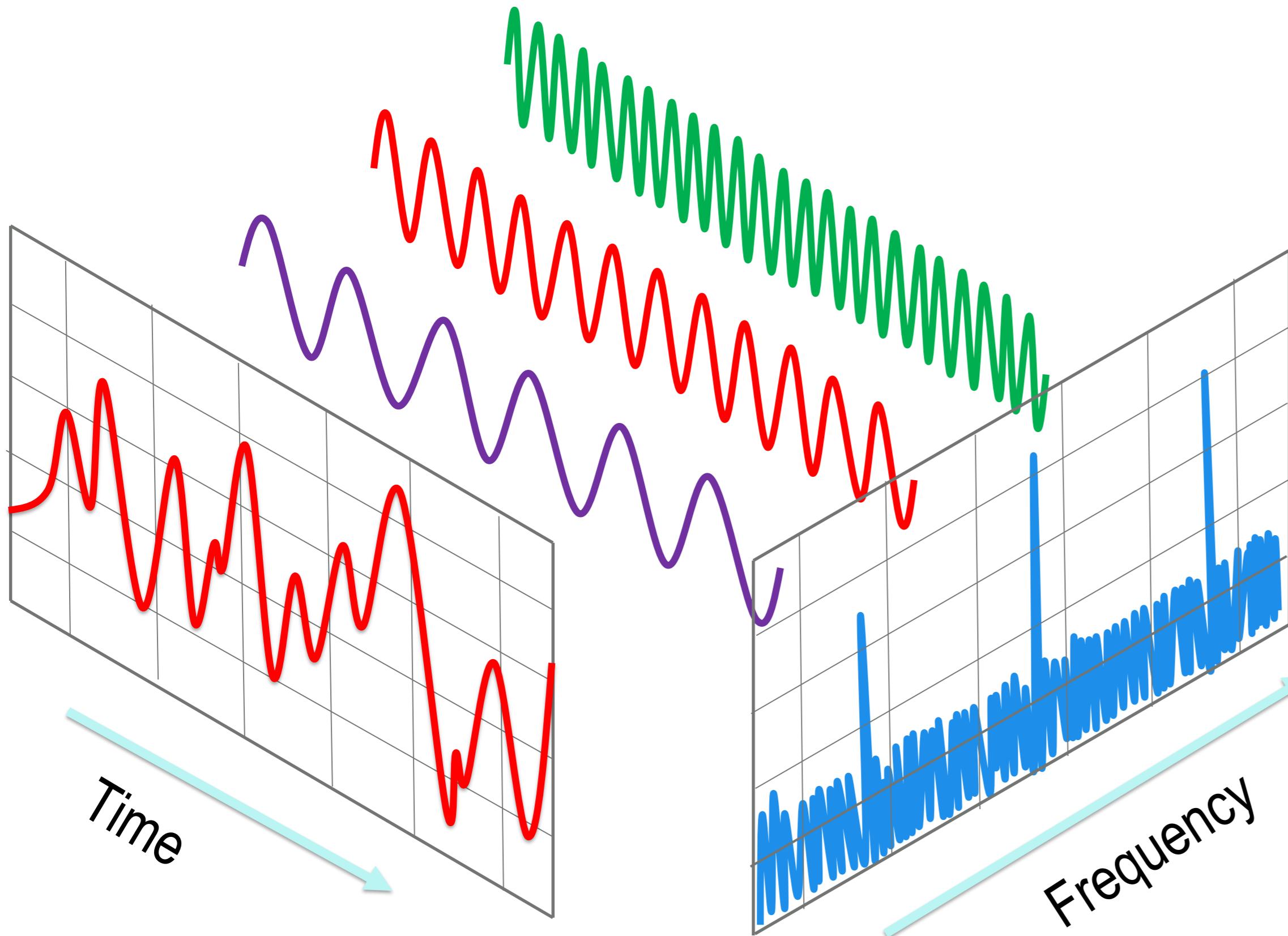
$$a_n = \frac{2}{T_o} \int_0^{T_o} x(t) \cos(nw_o t) dt = 0$$

$$b_n = \frac{2}{T_o} \int_0^{T_o} x(t) \sin(nw_o t) dt = \begin{cases} \frac{2}{n\pi} & n \text{ impar} \\ 0 & n \text{ par} \end{cases}$$



Señales – Recordatorio: Transformada de Fourier

Series de Fourier complejas – Empleando la formula de Euler $e^{\pm i\theta} = \cos\theta \pm i\sin\theta$

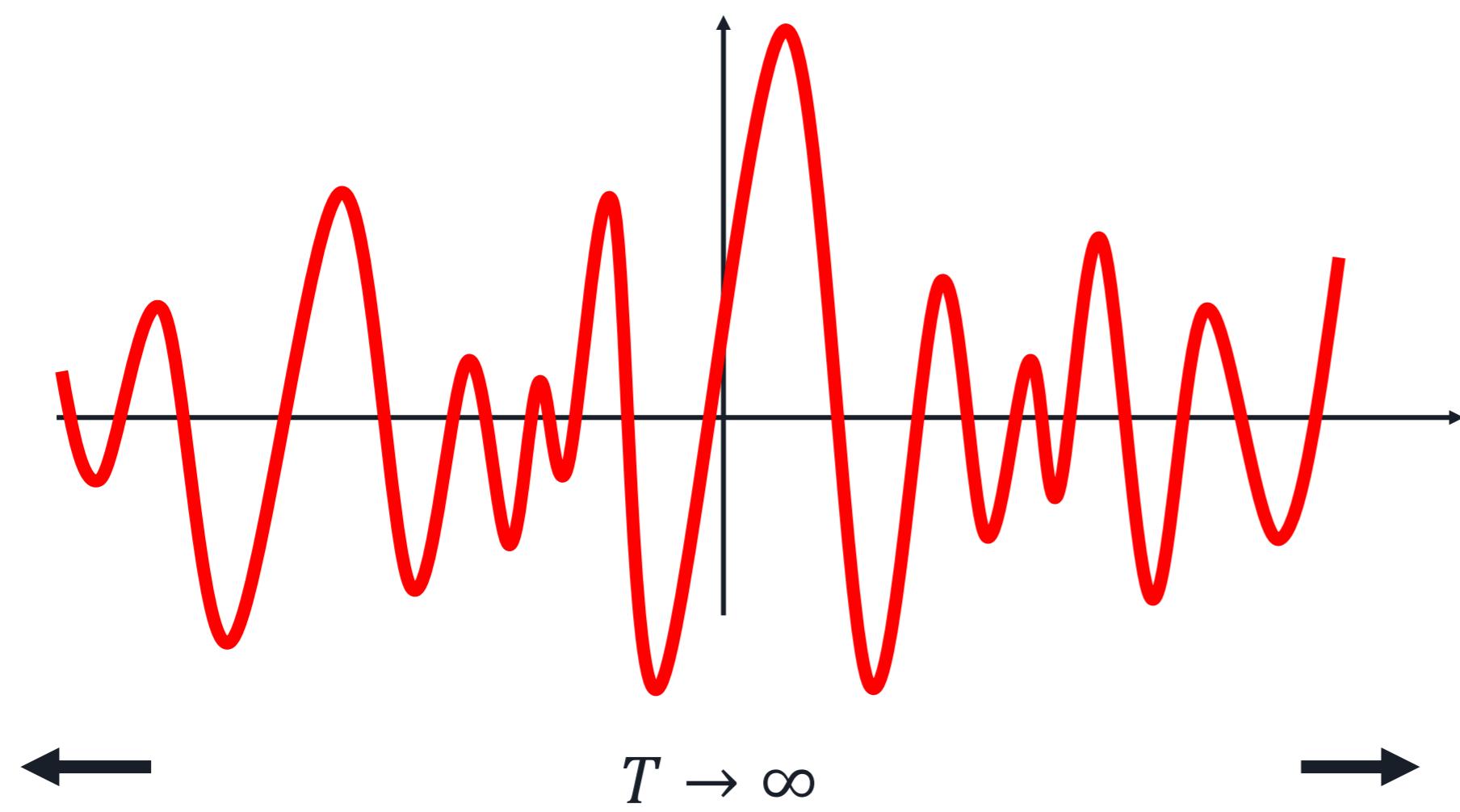


$$x(t) = \sum_{n=-\infty}^{\infty} X_n e^{inw_0 t}$$

$$X_n = \frac{1}{T_0} \int_0^{T_0} x(t) e^{-inw_0 t} dt$$

Señales – Recordatorio: Transformada de Fourier

Extensión para señales no periódicas



$$x(t) = \sum_{n=-\infty}^{\infty} X_n e^{i2\pi n f_o t}$$

$$X_n = \frac{1}{T_0} \int_0^{T_0} x(t) e^{-inw_o t} dt$$

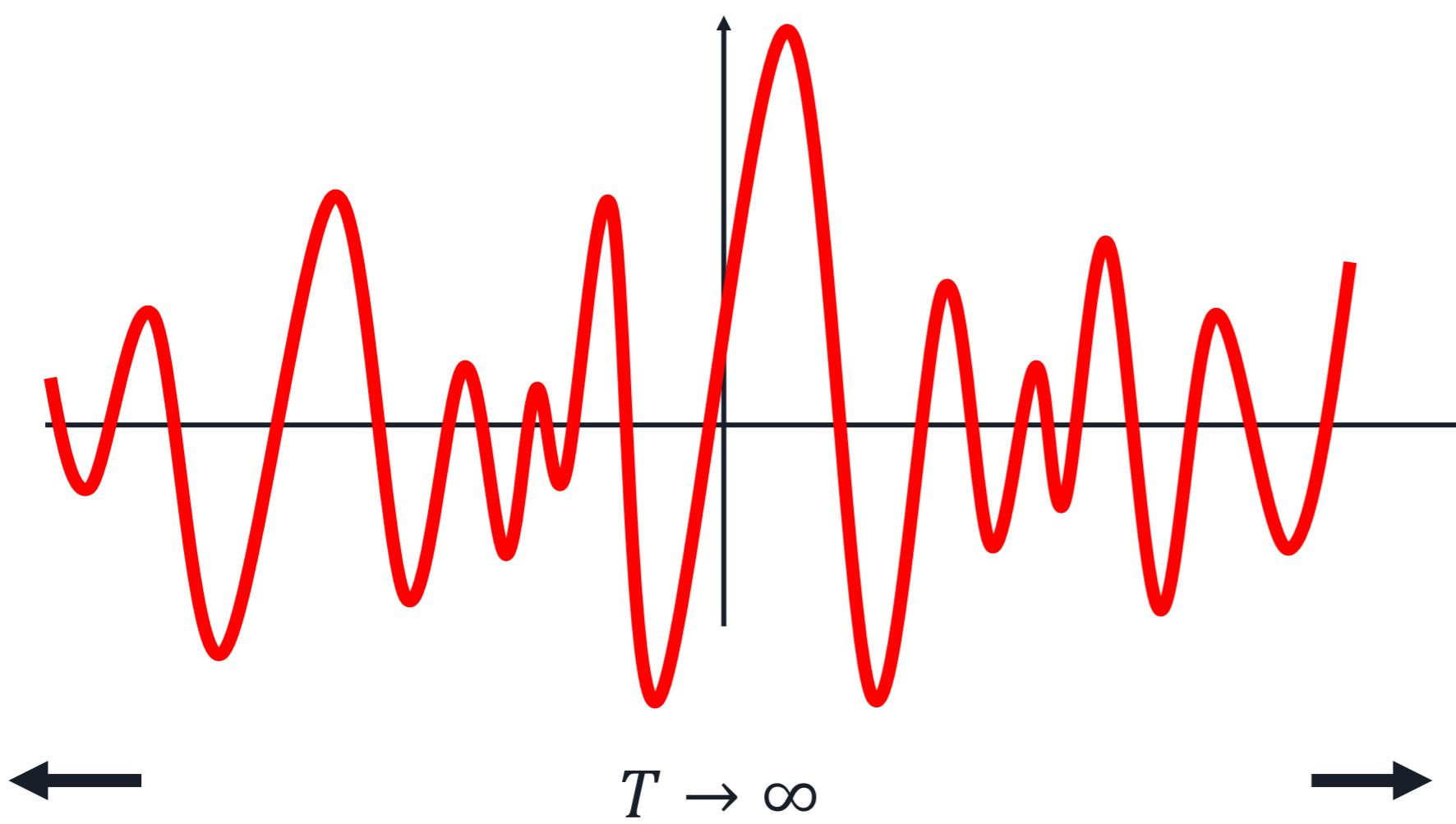
$$x(t) = \sum_{n=-\infty}^{\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(\xi) e^{i2\pi n f_o (t-\xi)} d\xi$$

$$T \rightarrow \infty \quad \frac{1}{T} \rightarrow df$$

$$x(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(\tau) e^{i2\pi f(t-\tau)} d\tau df$$

Señales – Recordatorio: Transformada de Fourier

Extensión para señales no periódicas



**Fourier
Transform:**

Inverse FT:

$$\hat{x}(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi f t} dt$$

$$x(t) = \int_{-\infty}^{\infty} \hat{x}(f) e^{i2\pi f t} df$$

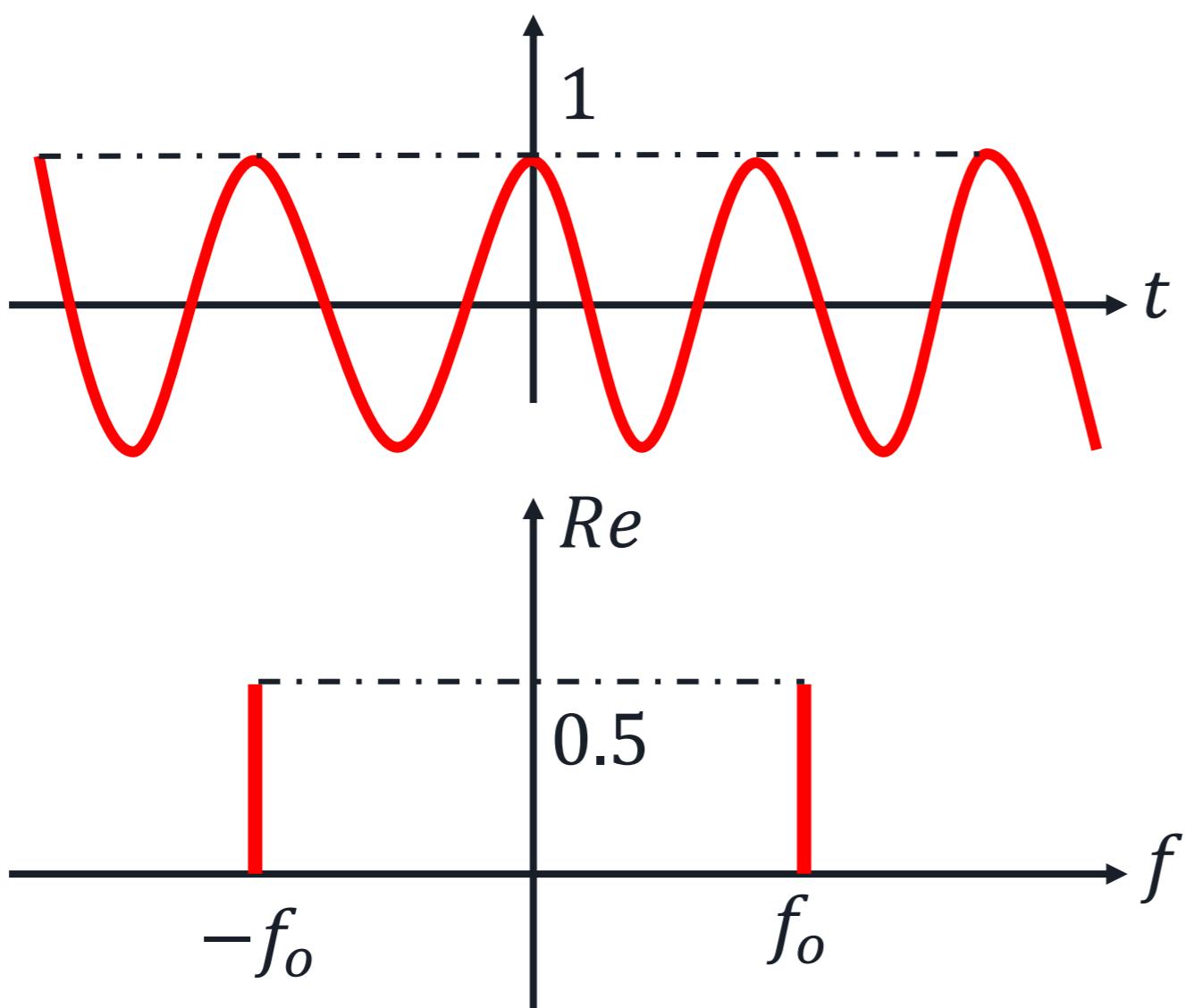


Señales – Recordatorio: Transformada de Fourier

Algunos casos comunes:

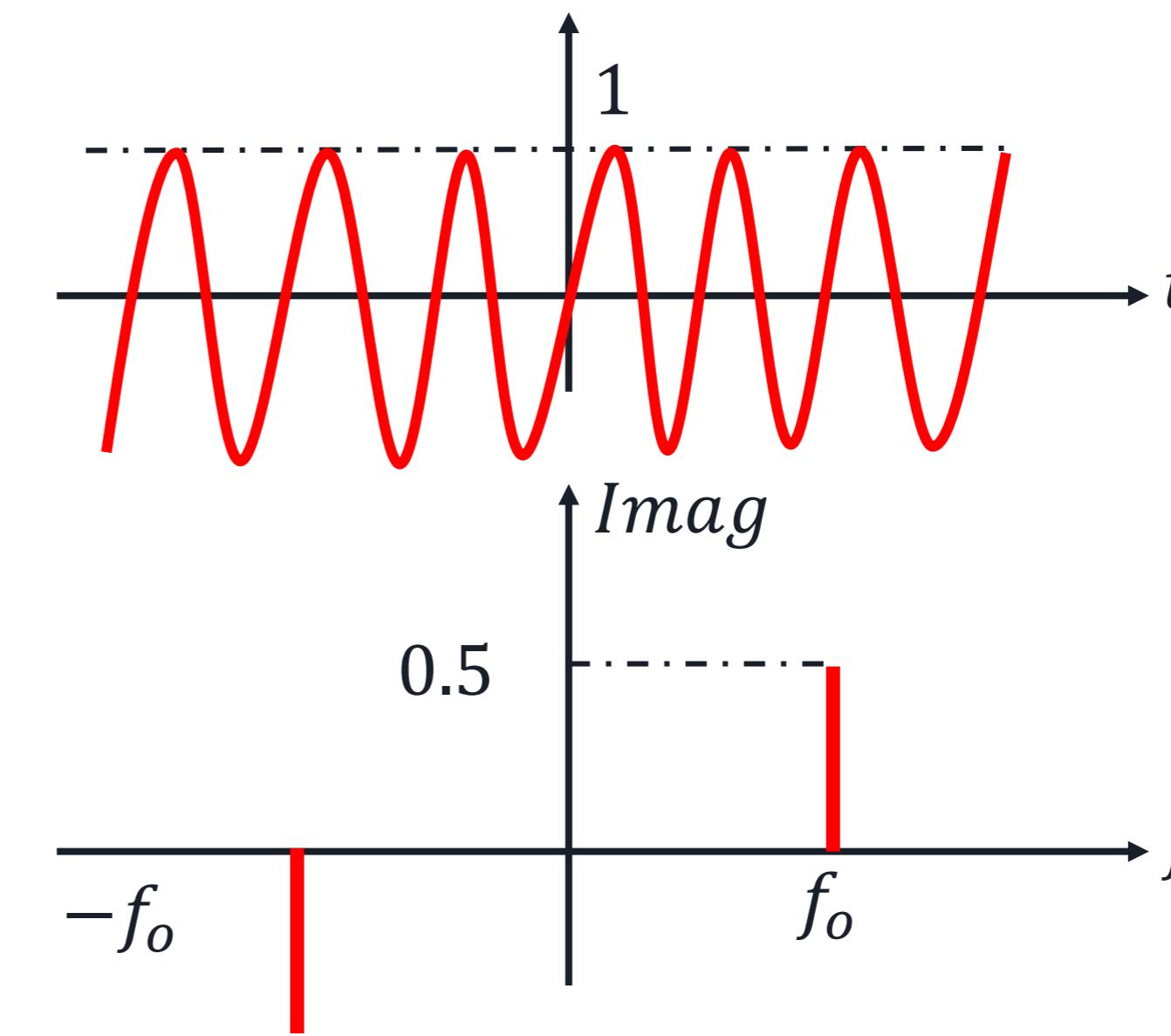
- Función coseno:

$$\mathcal{F}(\cos(2\pi f_0 t)) = \frac{1}{2}[-\delta(f - f_0) + \delta(f + f_0)]$$



- Función seno:

$$\mathcal{F}(\sin(2\pi f_0 t)) = \frac{i}{2}[-\delta(f - f_0) + \delta(f + f_0)]$$



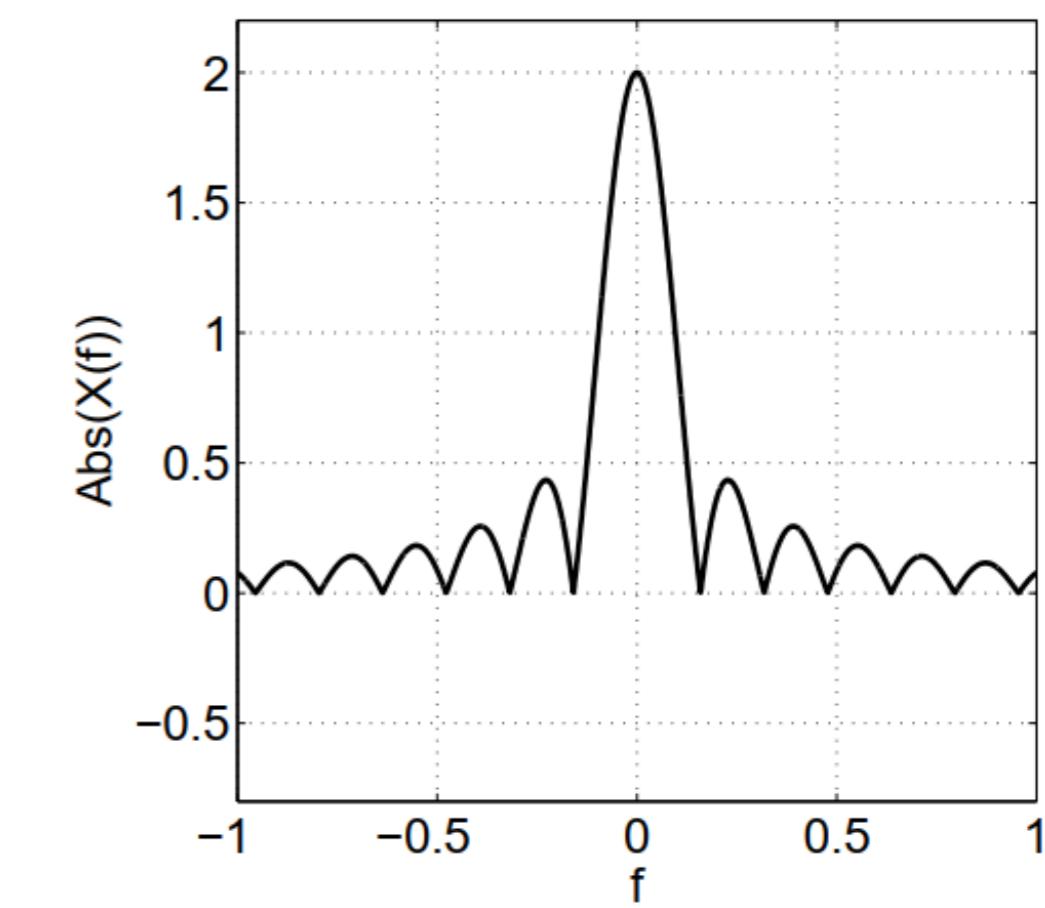
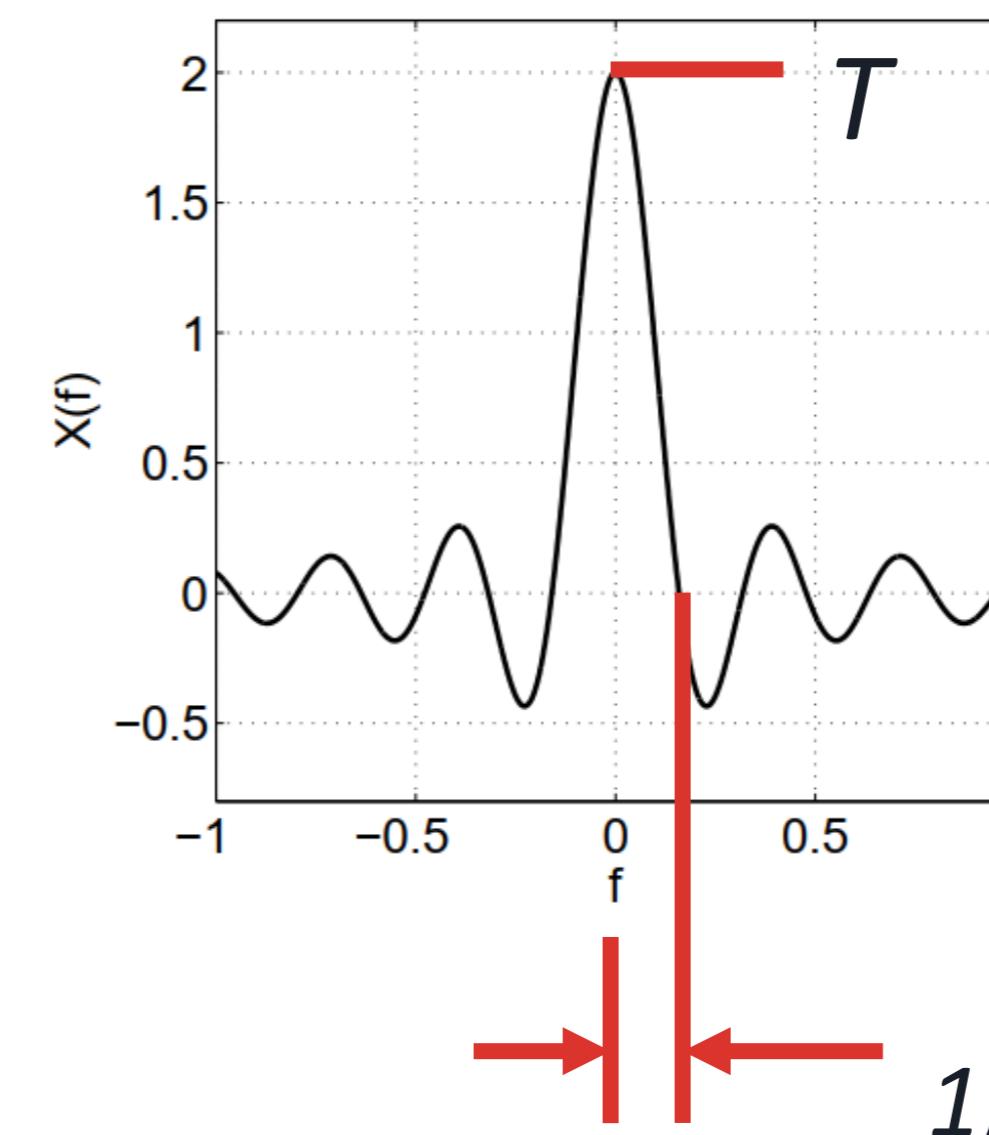
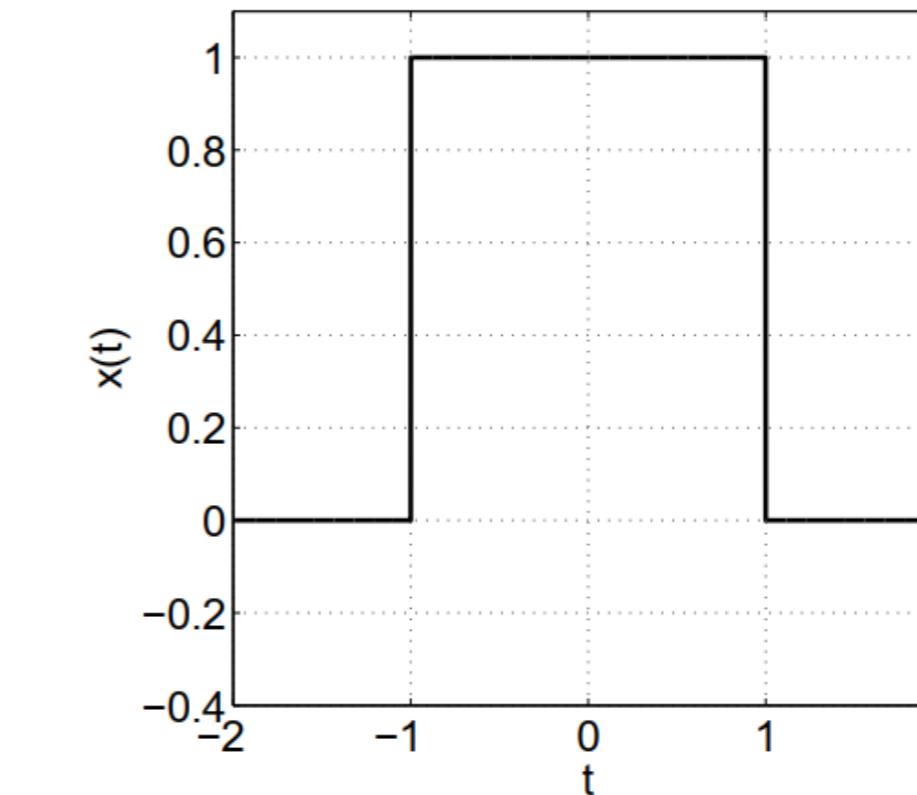
Señales – Recordatorio: Transformada de Fourier

Algunos casos comunes:

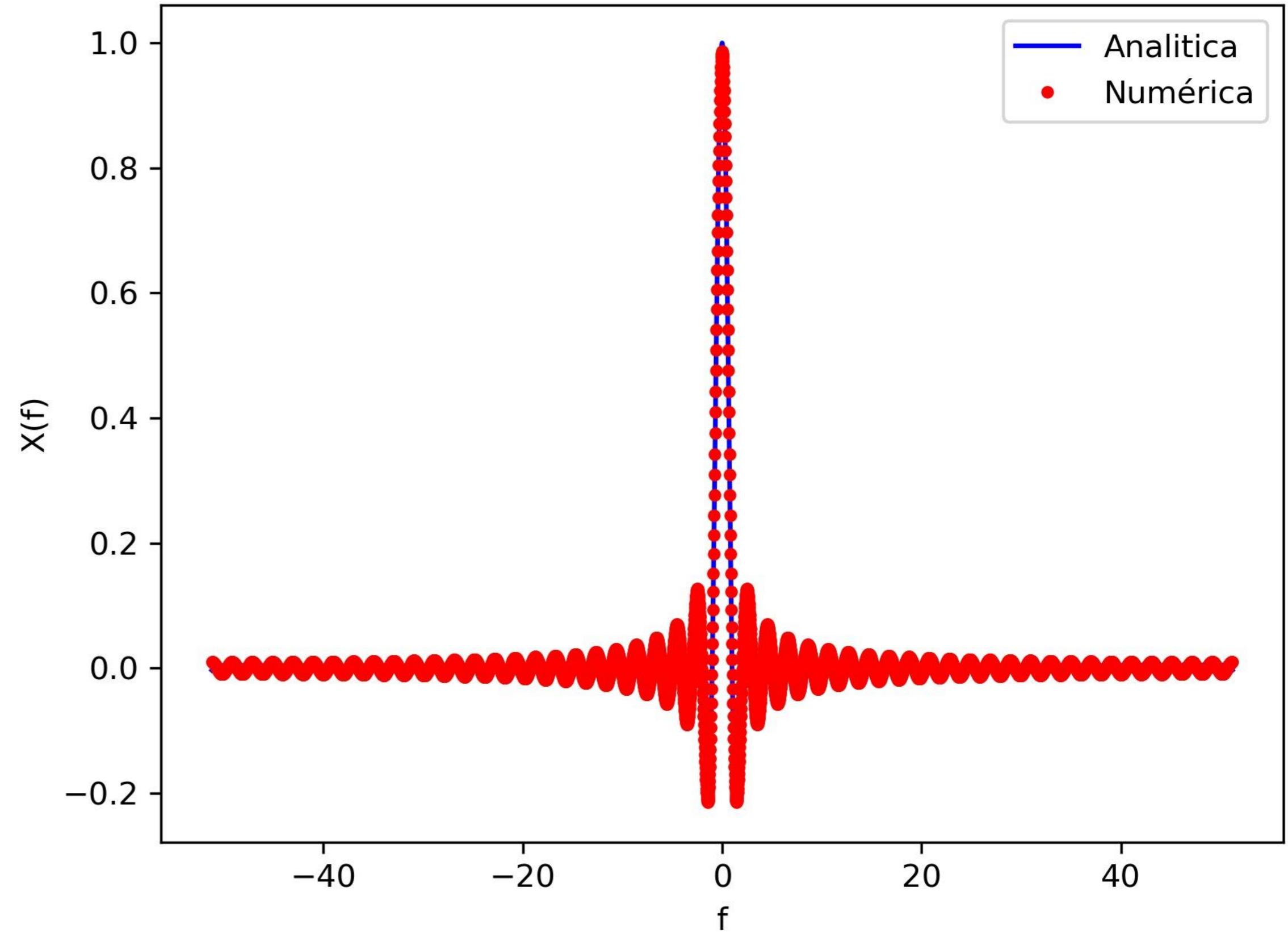
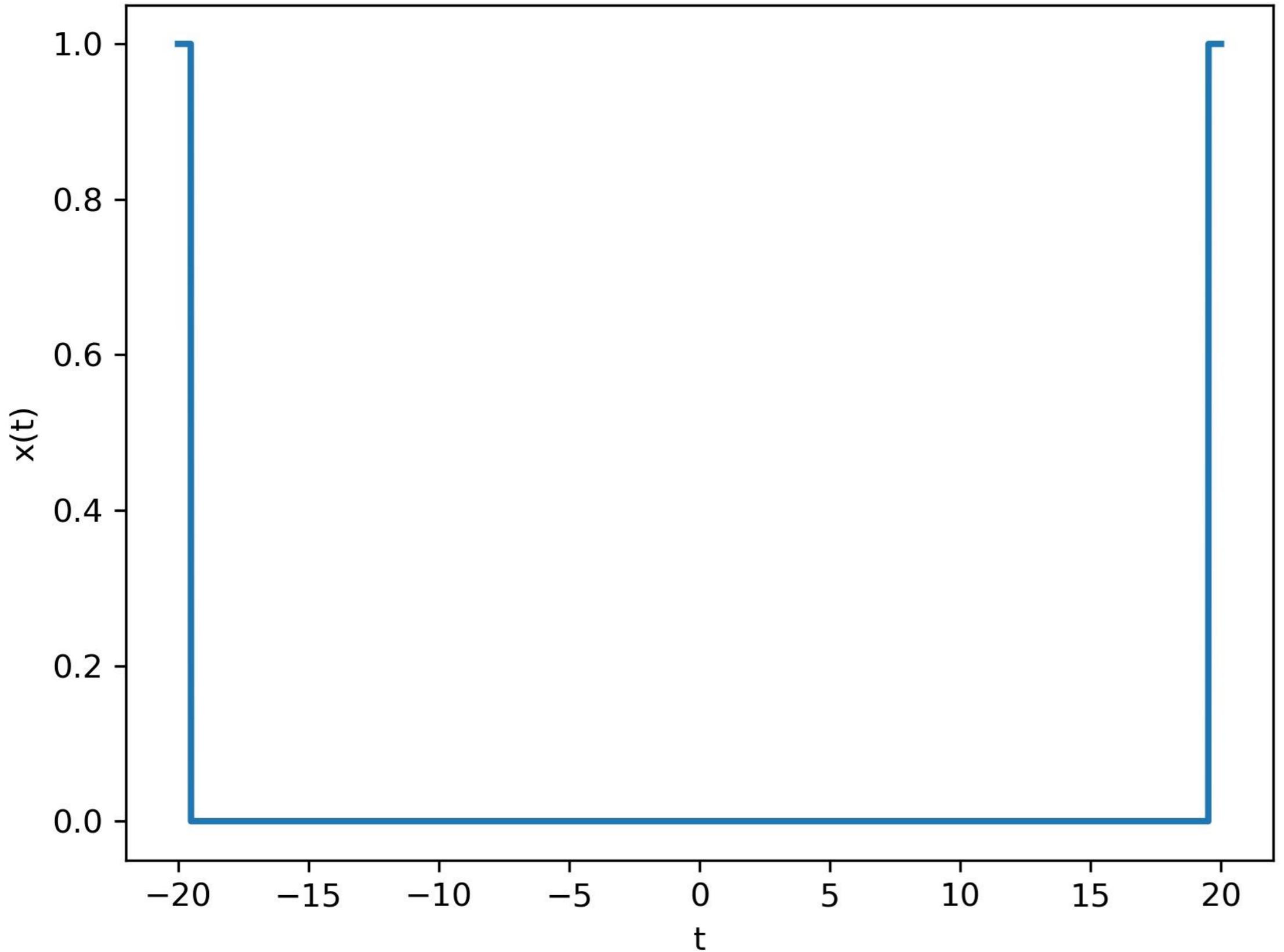
- Box function:

$$b_T(t) = \begin{cases} 1 & t \in [-T/2, T/2] \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned}\hat{b}_T(f) &= \int_{-T/2}^{T/2} e^{-i2\pi f t} dt = \frac{e^{-i2\pi f T} - e^{i2\pi f T}}{i2\pi f} = \\ &= T \frac{\sin(\pi f T)}{\pi f T} = T \operatorname{sinc}(\pi f T)\end{aligned}$$



Señales – Recordatorio: Transformada de Fourier



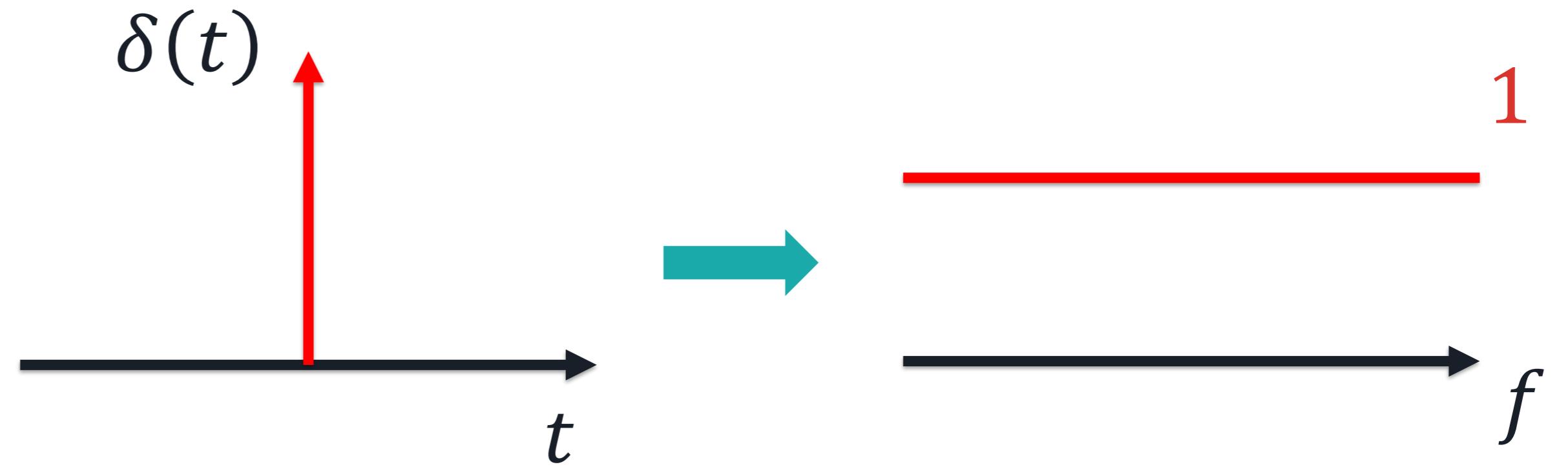
Señales – Recordatorio: Transformada de Fourier

Algunos casos comunes:

- Función delta de Dirac:

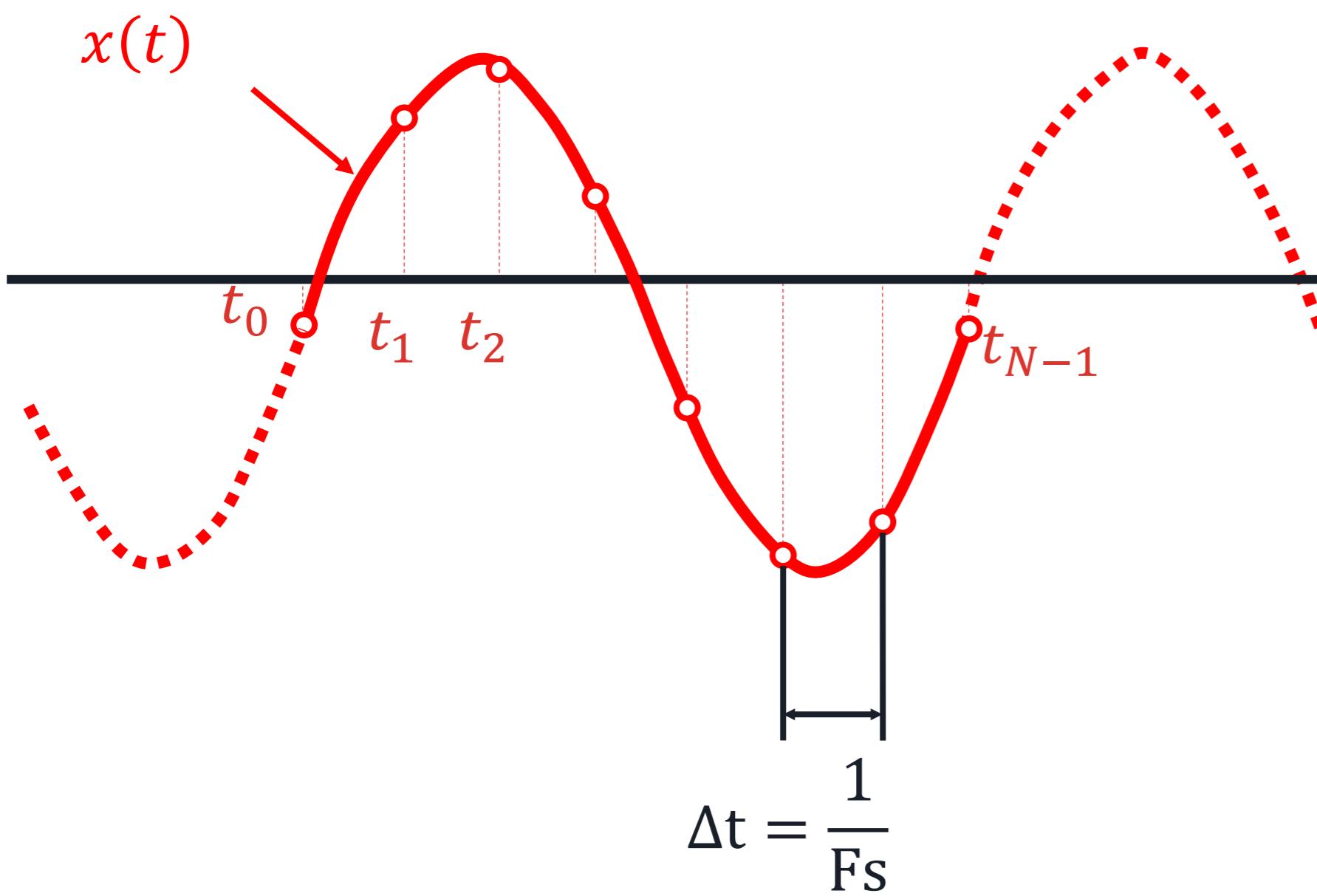
$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases}$$

$$\hat{\delta}(f) = \int_{-\infty}^{\infty} \delta(t) e^{-i2\pi f t} dt = 1$$



Señales – Recordatorio: Transformada de Fourier

Transformada de Fourier discreta:



Discrete Fourier Transform (DFT)

$$F(\omega) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t} dt = f(t_0)e^{-i\omega t_0} + \dots + f(t_{N-1})e^{-i\omega t_{N-1}}$$

$$F(\omega) = \sum_{k=0}^{N-1} x(t_k)e^{-i\omega(t_k)} = \sum_{k=0}^{N-1} x(t_0 + k\Delta t)e^{-i\omega(t_0 + k\Delta t)}$$

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi}{N}kn} \quad x_n = \frac{1}{N} \sum_{n=0}^{N-1} X_k e^{\frac{i2\pi}{N}kn} \quad n = 0, \dots, N - 1$$

Fast Fourier Transform (FFT)

El algoritmo FFT se emplea para obtener la DFT mediante la factorización de la matriz DFT en el producto de factores de pequeñas dimensiones (sparse factors).

Coste computacional: $O(N^2) \rightarrow O(N \log N)$

Para mayor información:

Cetin, A. E. (2012). Lecture notes on discrete-time signal processing.

<http://kilyos.ee.bilkent.edu.tr/~ee424/Lecture%20Slides/Lecture%20Notes/Chapter1.pdf>

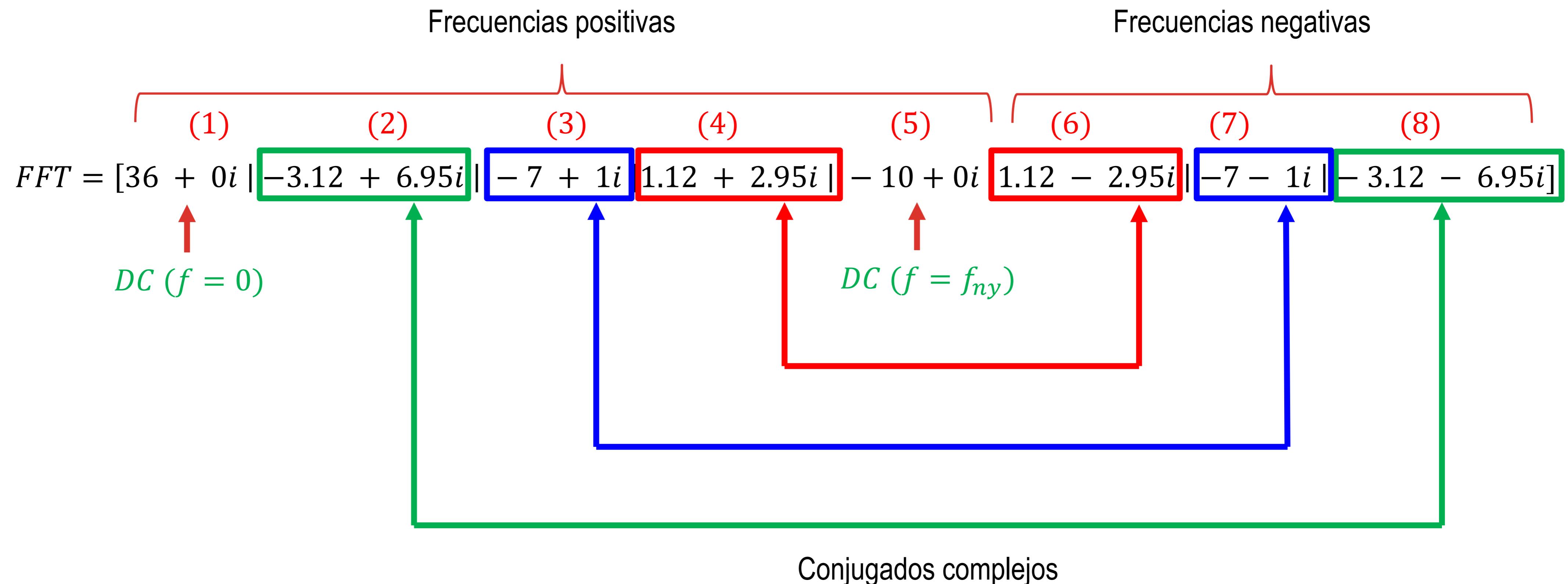


Señales – Recordatorio: Transformada de Fourier

Algoritmo FFT

$$x = [1|3|4|5|2|8|6|7]$$

$$FFT = fft(x)$$

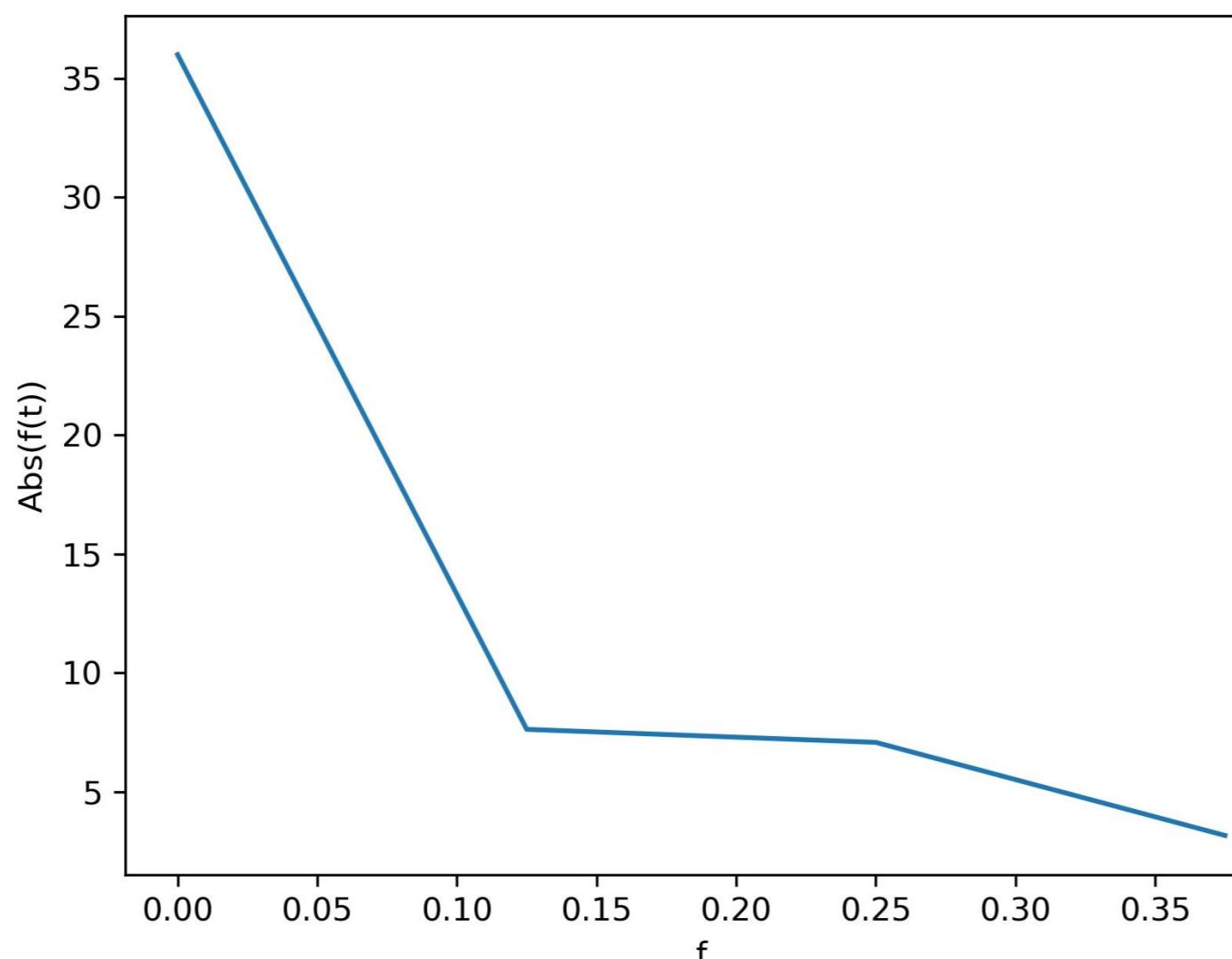


Señales – Recordatorio: Transformada de Fourier

FFT en Python

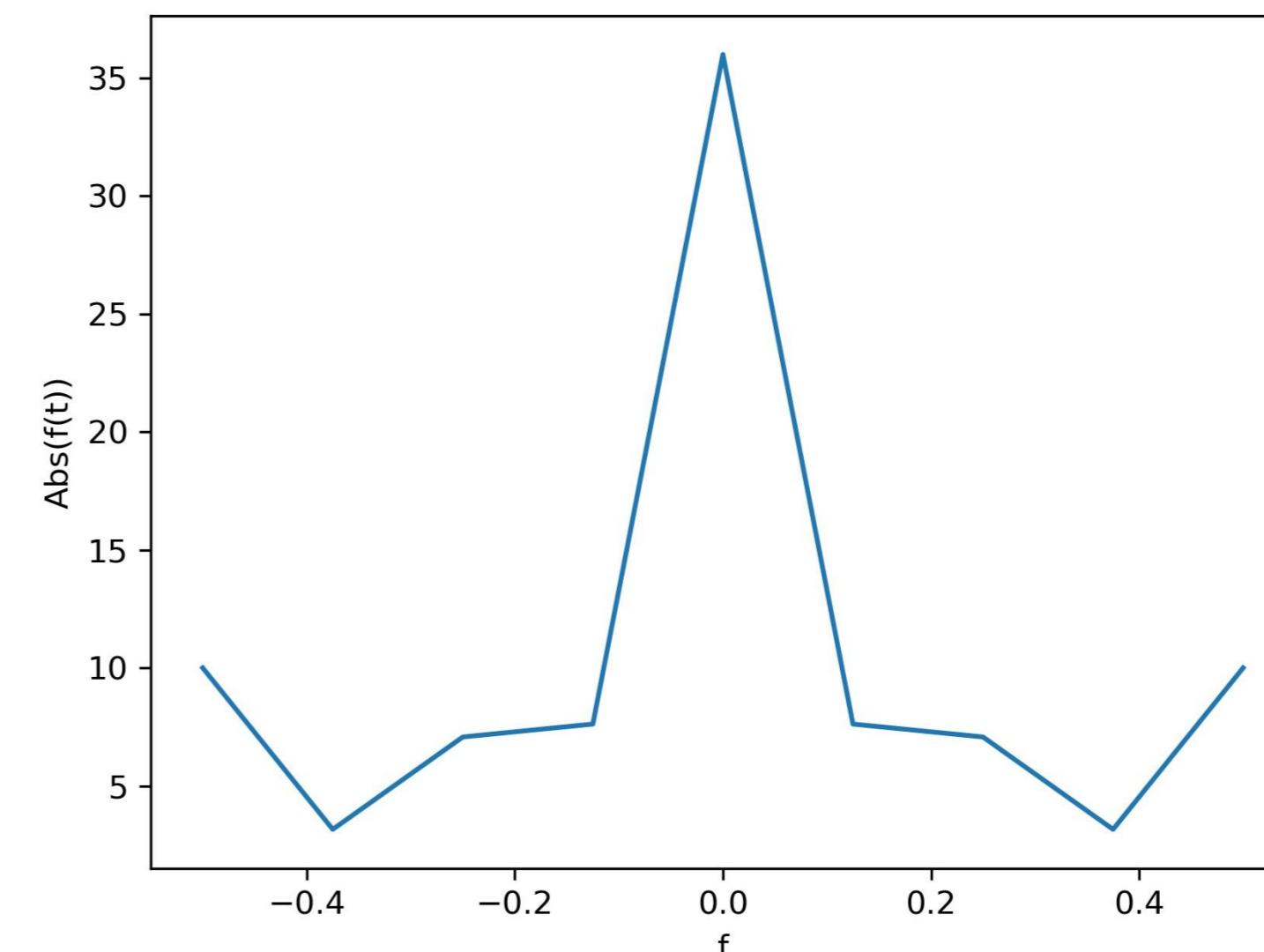
Espectro de una hoja

```
x = np.array([1,3,4,5,2,8,6,7])
N = len(x)
Nmed = int(N/2)
FFTr = fft(x)
Mod = np.abs(FFTr)
Po = Mod[0:Nmed]
f = np.arange(0,Nmed)*(Fs/N)
```



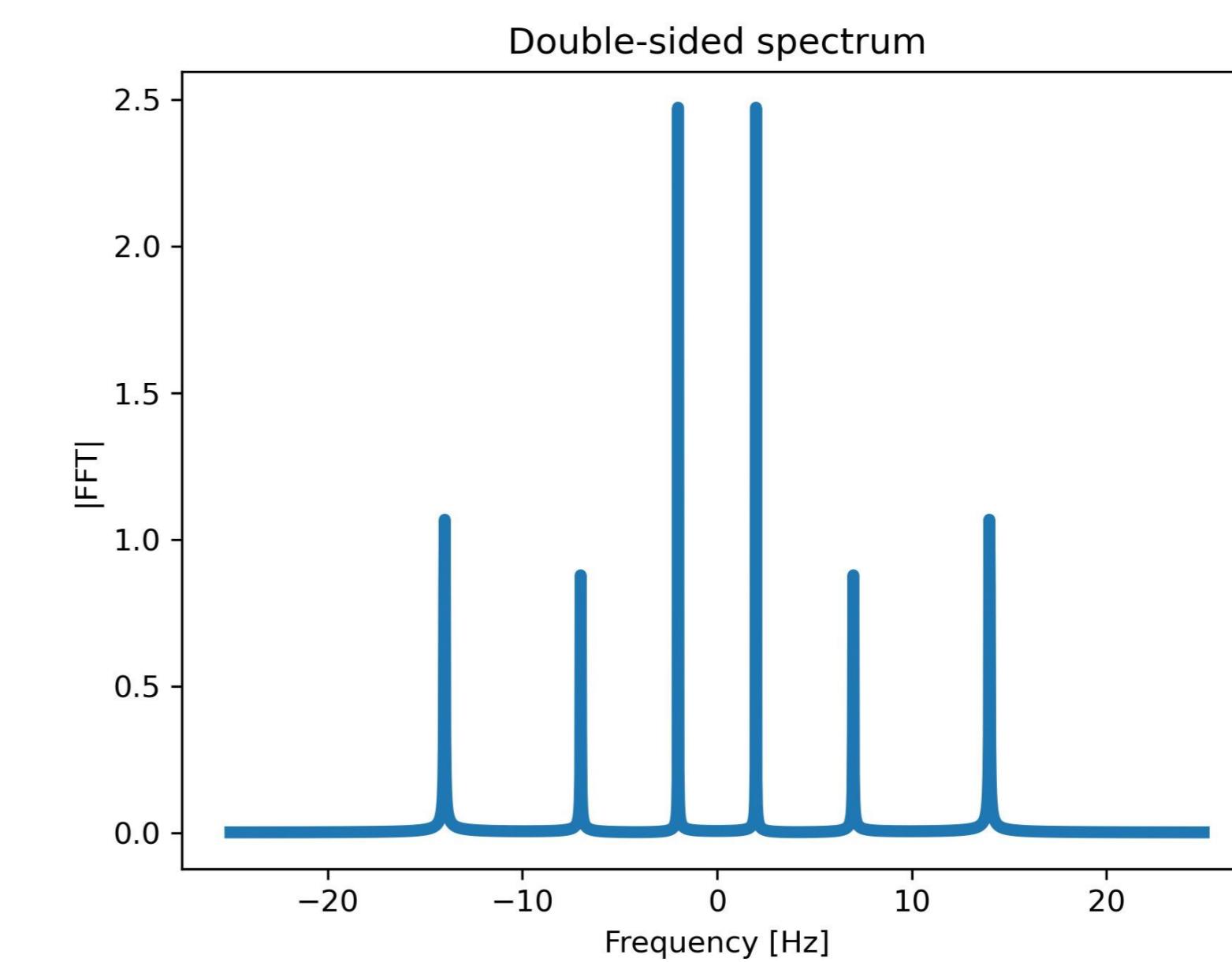
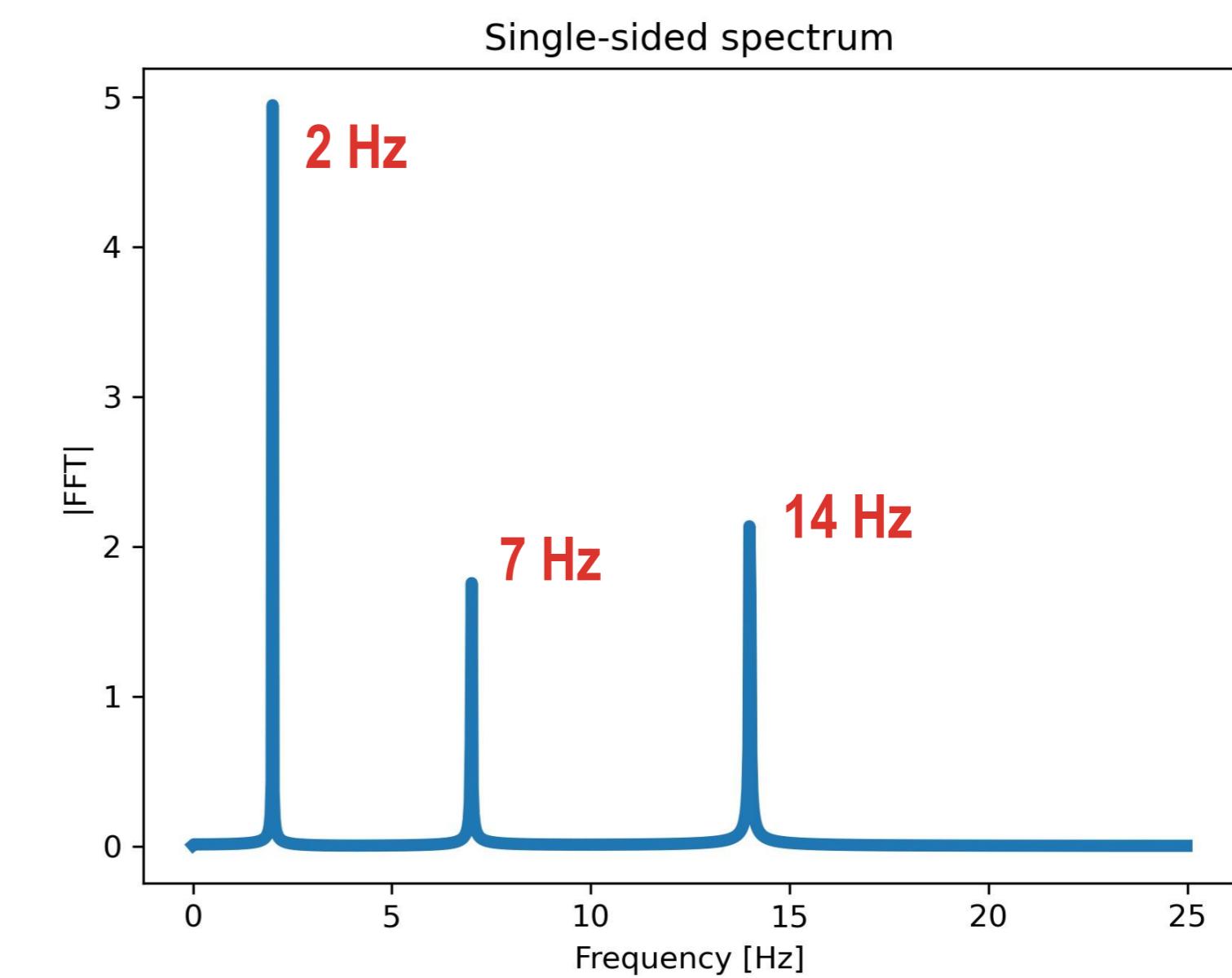
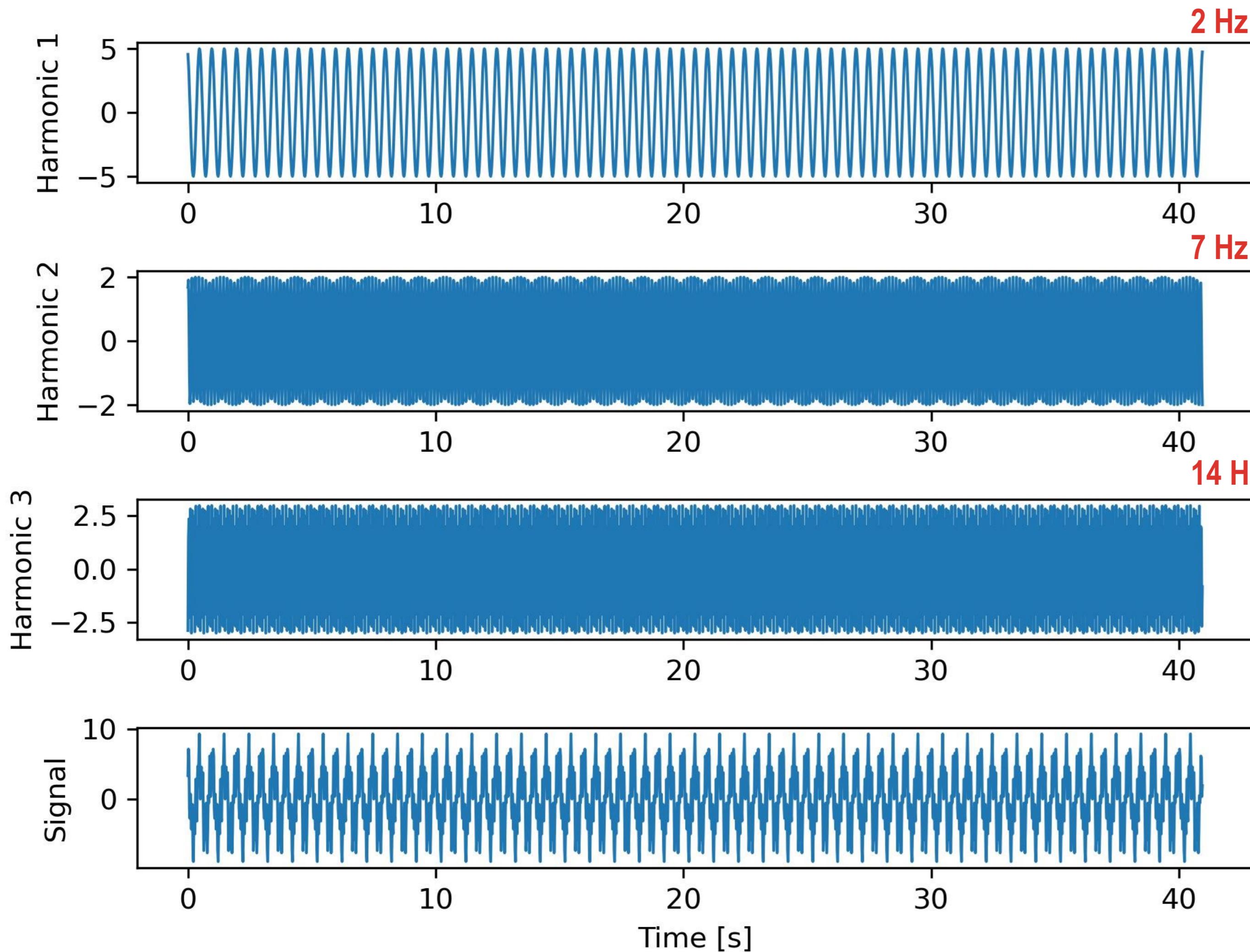
Espectro de doble hoja

```
f = np.arange(-Nmed,Nmed+1)*(Fs/N)
Pt = np.concatenate((Mod[Nmed:],Mod[0:Nmed]))
```



Señales – Recordatorio: Transformada de Fourier

Ejemplo señales armónicas



Señales – Recordatorio: Transformada de Fourier

Teorema de Parseval

Si $x[n]$ y $v[n]$ son señales de valores reales:

$$\sum_{n=-\infty}^{\infty} x[n]v[n] = \frac{1}{2\pi} \int_0^{2\pi} \overline{X(\Omega)}V(\Omega) d\Omega$$

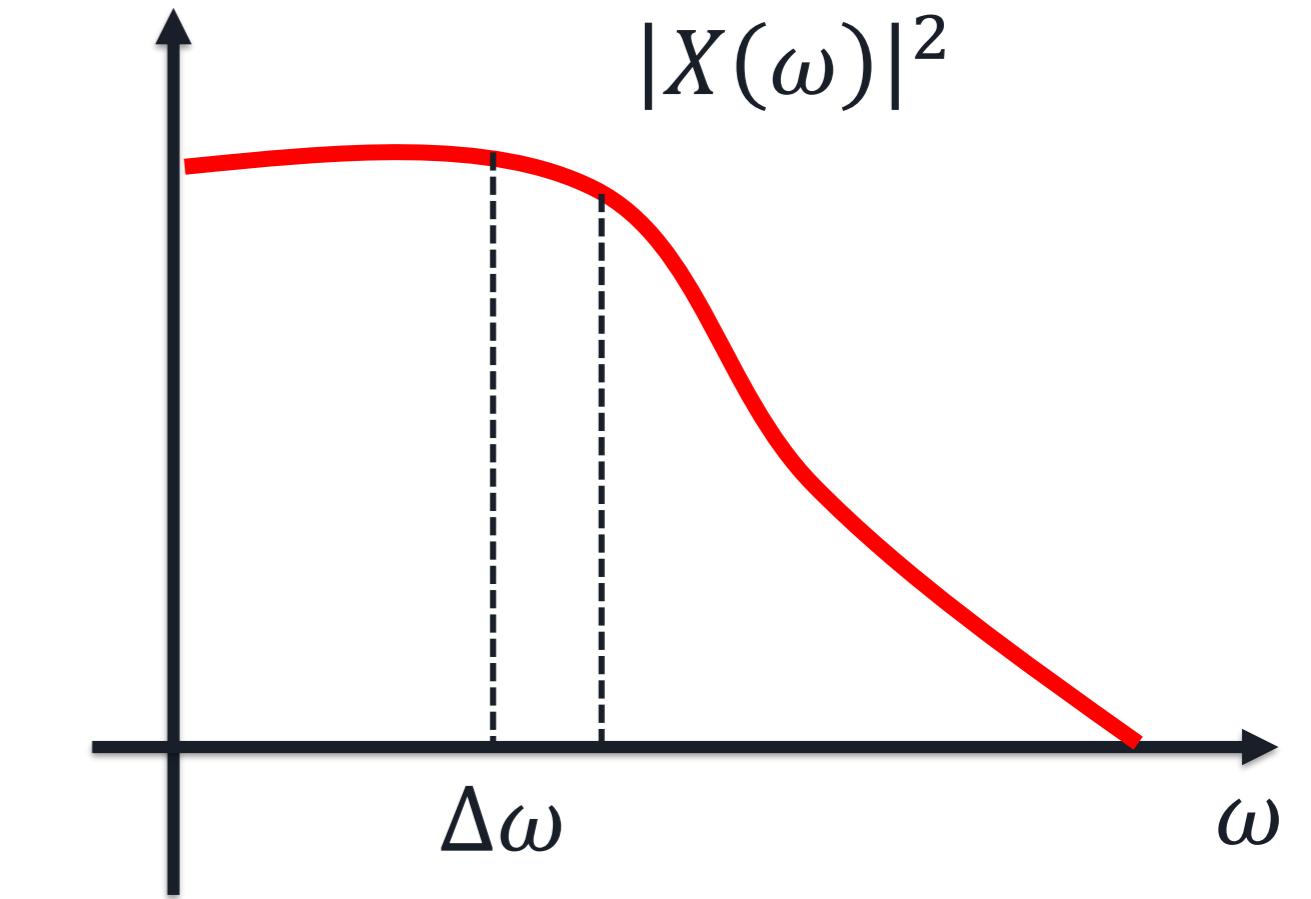
La energía de la señal se puede obtener como:

$$\sum_{n=-\infty}^{\infty} x^2[n]$$

la cual puede también determinarse en el dominio de la frecuencia como:

$$\sum_{n=-\infty}^{\infty} x^2[n] = \frac{1}{2\pi} \int_0^{2\pi} |X(\Omega)|^2 d\Omega \quad \text{y para una señal discreta}$$

$$\sum_{n=0}^{N-1} (x_n)^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2$$



↑
Energy Spectral Density



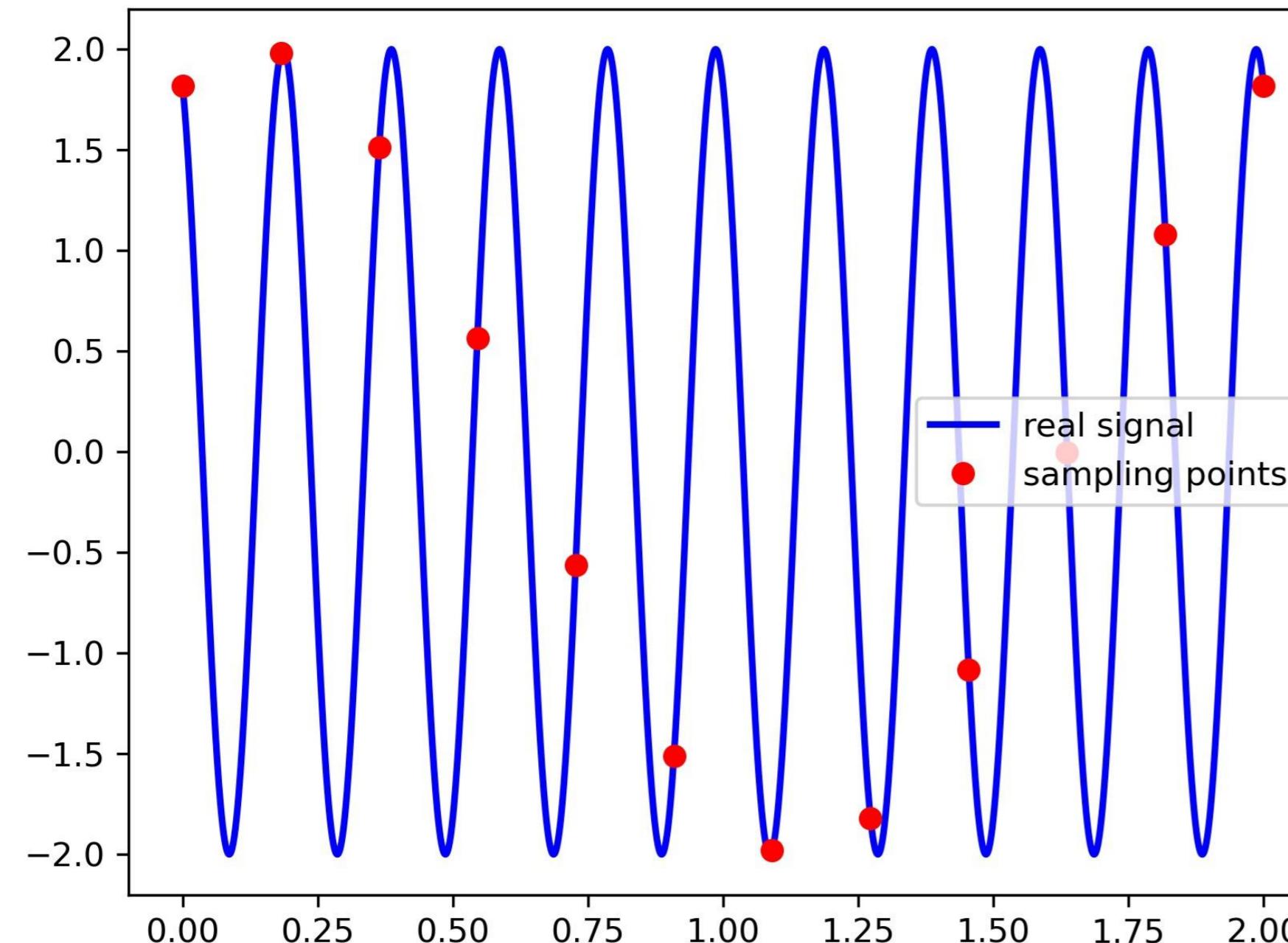
UNIVERSIDAD
DE GRANADA



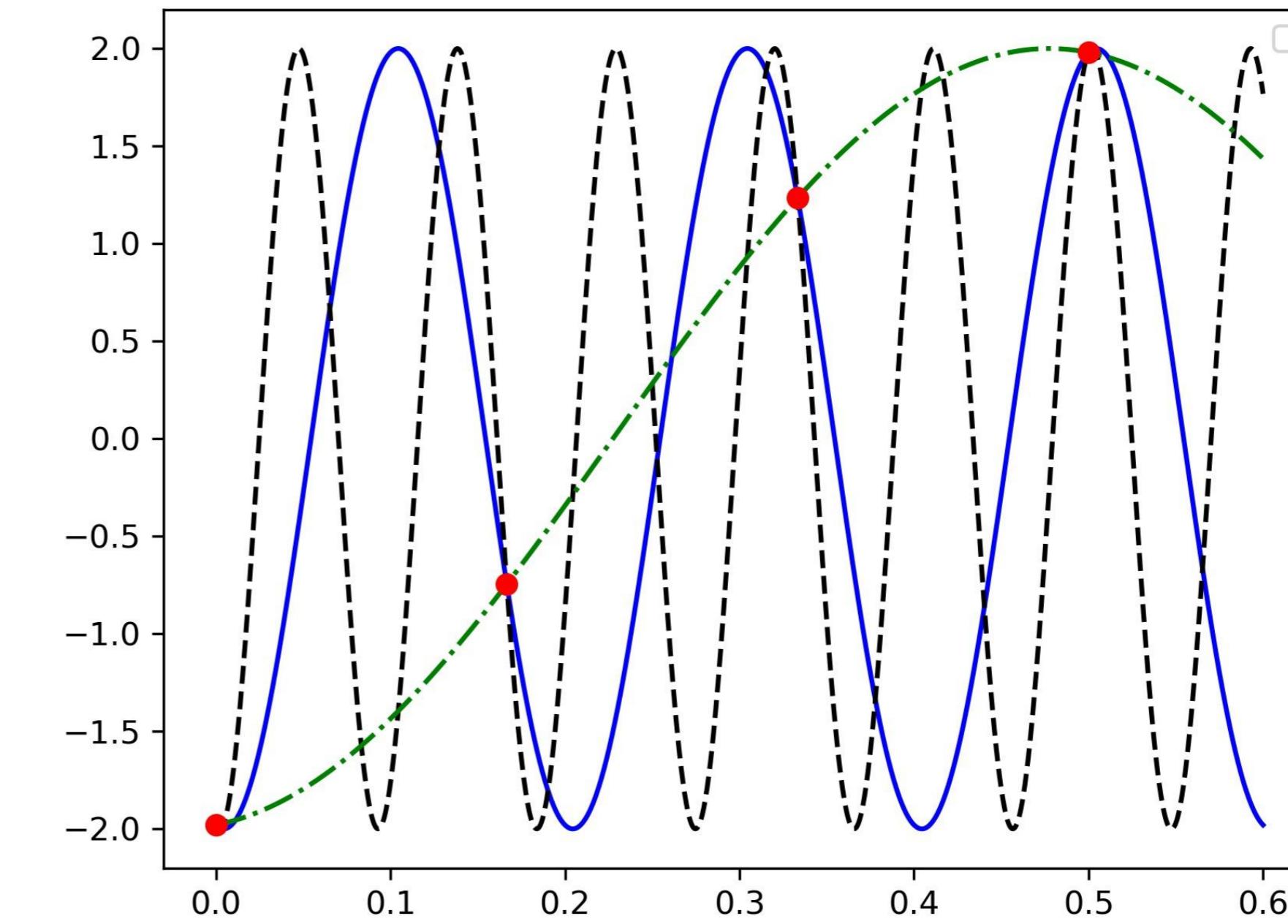
Aliasing.

Aliasing

El aliasing es un efecto que hace que diferentes señales se vuelvan indistinguibles cuando se muestran.



[F_Aliasing_spline.py](#)

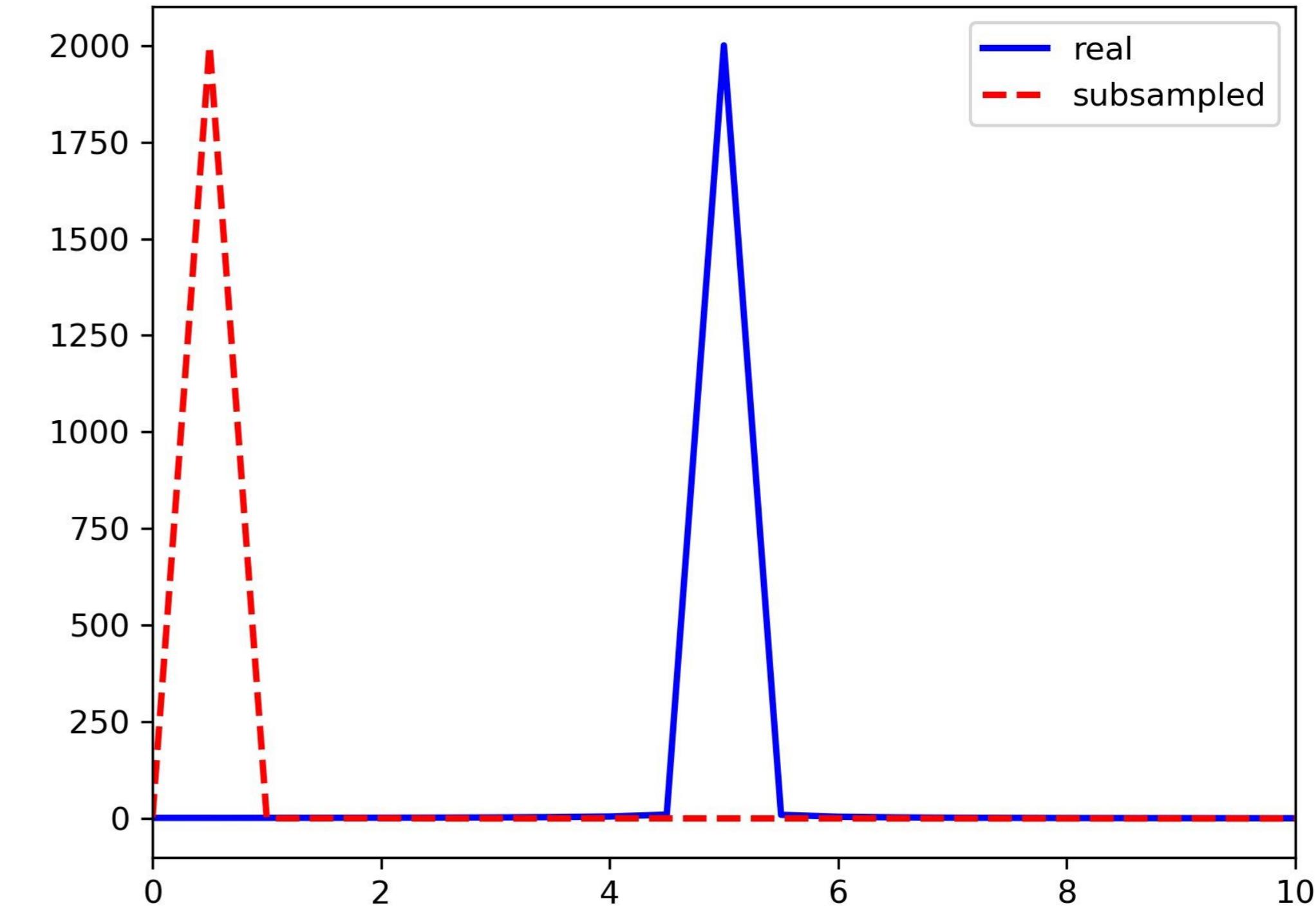
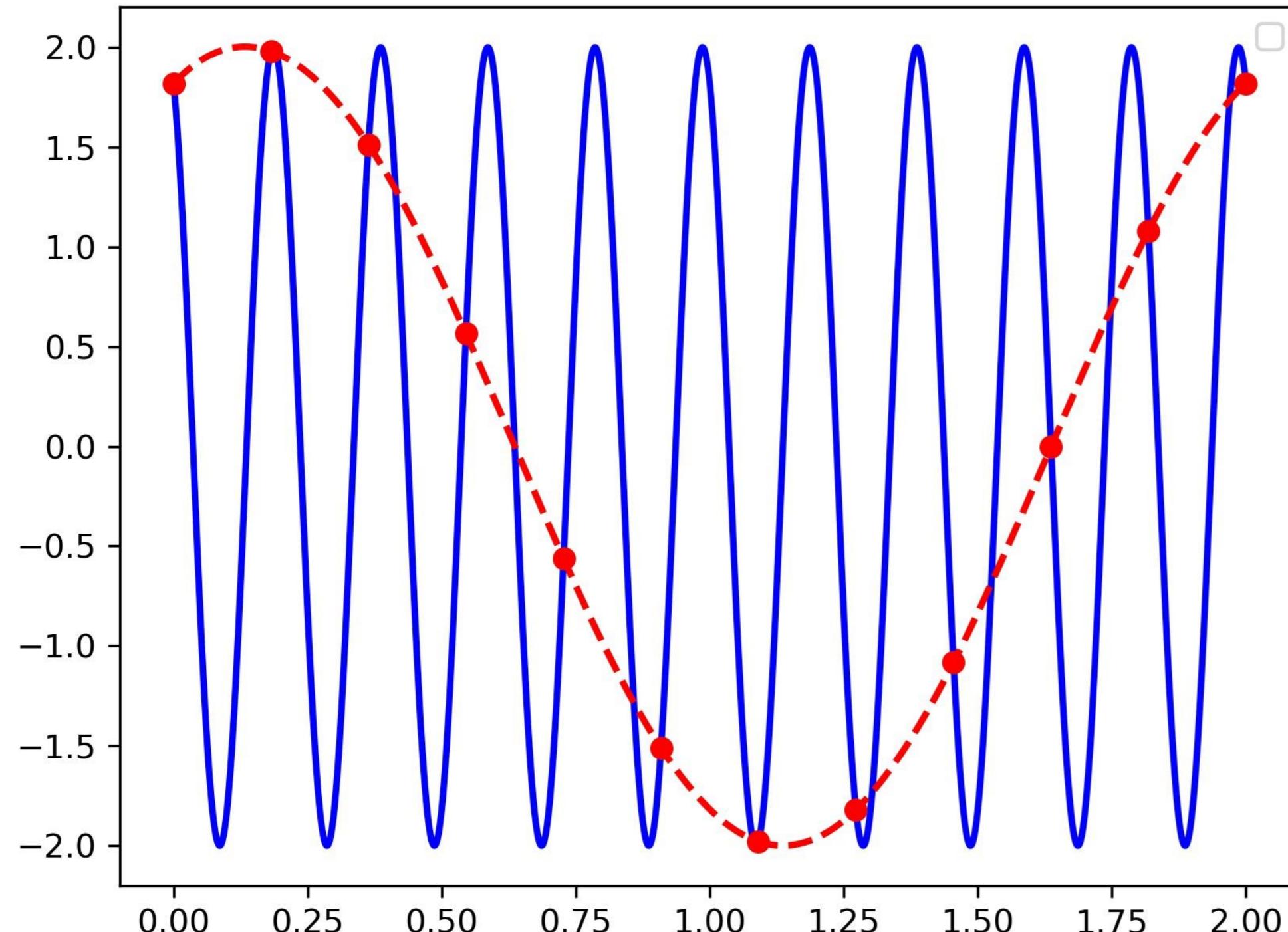


[E_Aliasing.py](#)



Aliasing

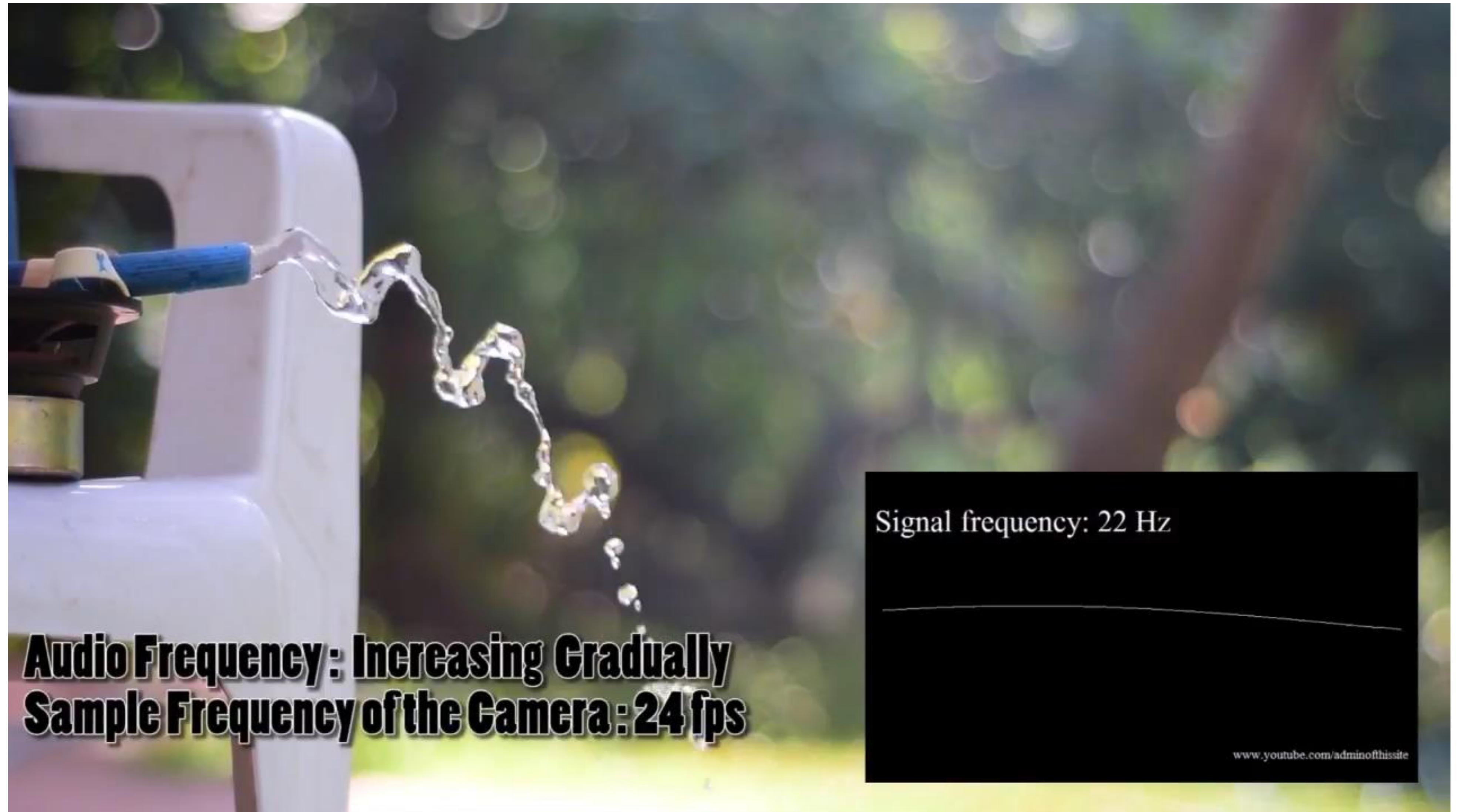
El **aliasing** es un efecto que hace que diferentes señales se vuelvan indistinguibles cuando se muestran.



[F_Aliasing_spline.py](#)



Aliasing



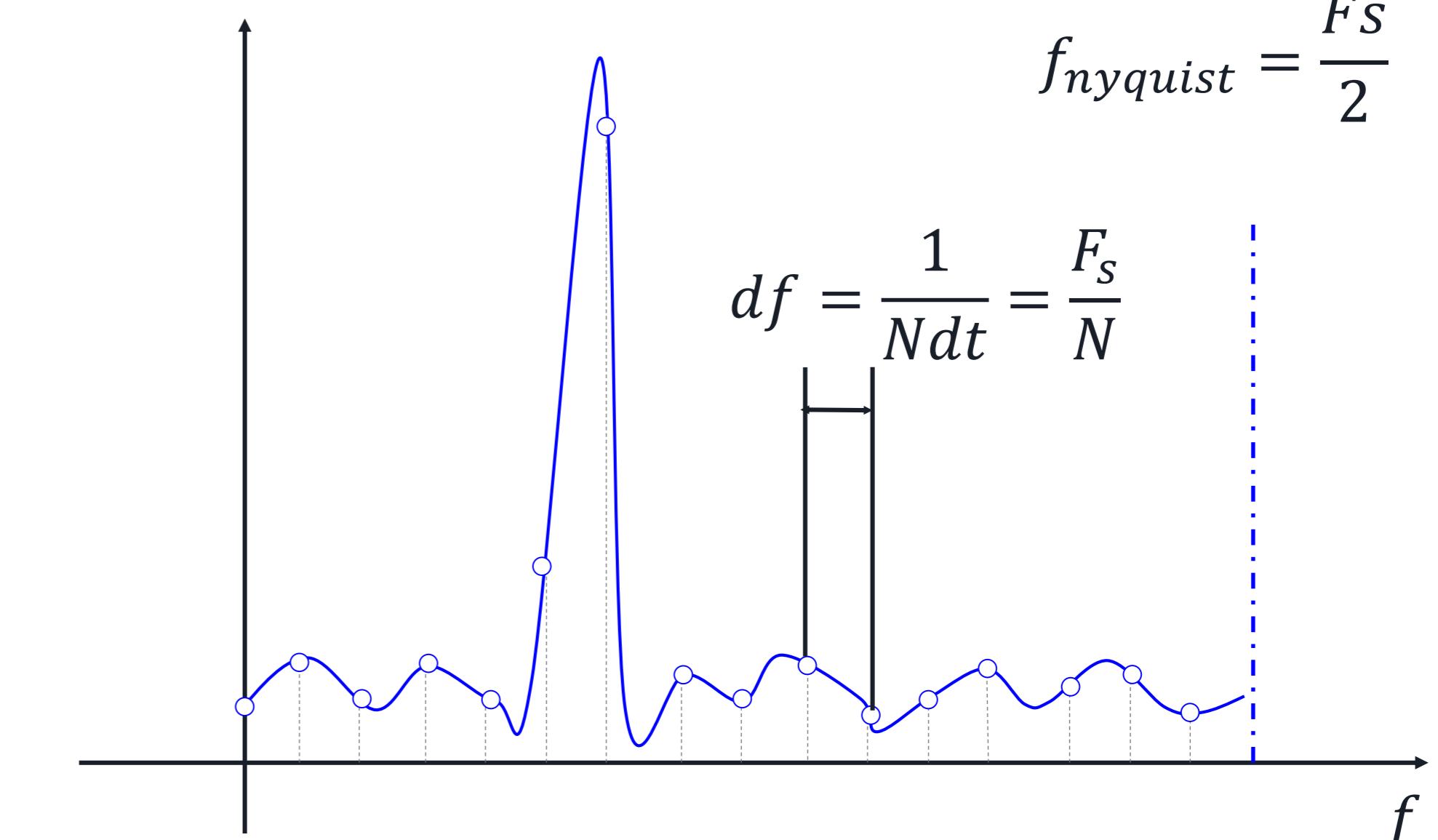
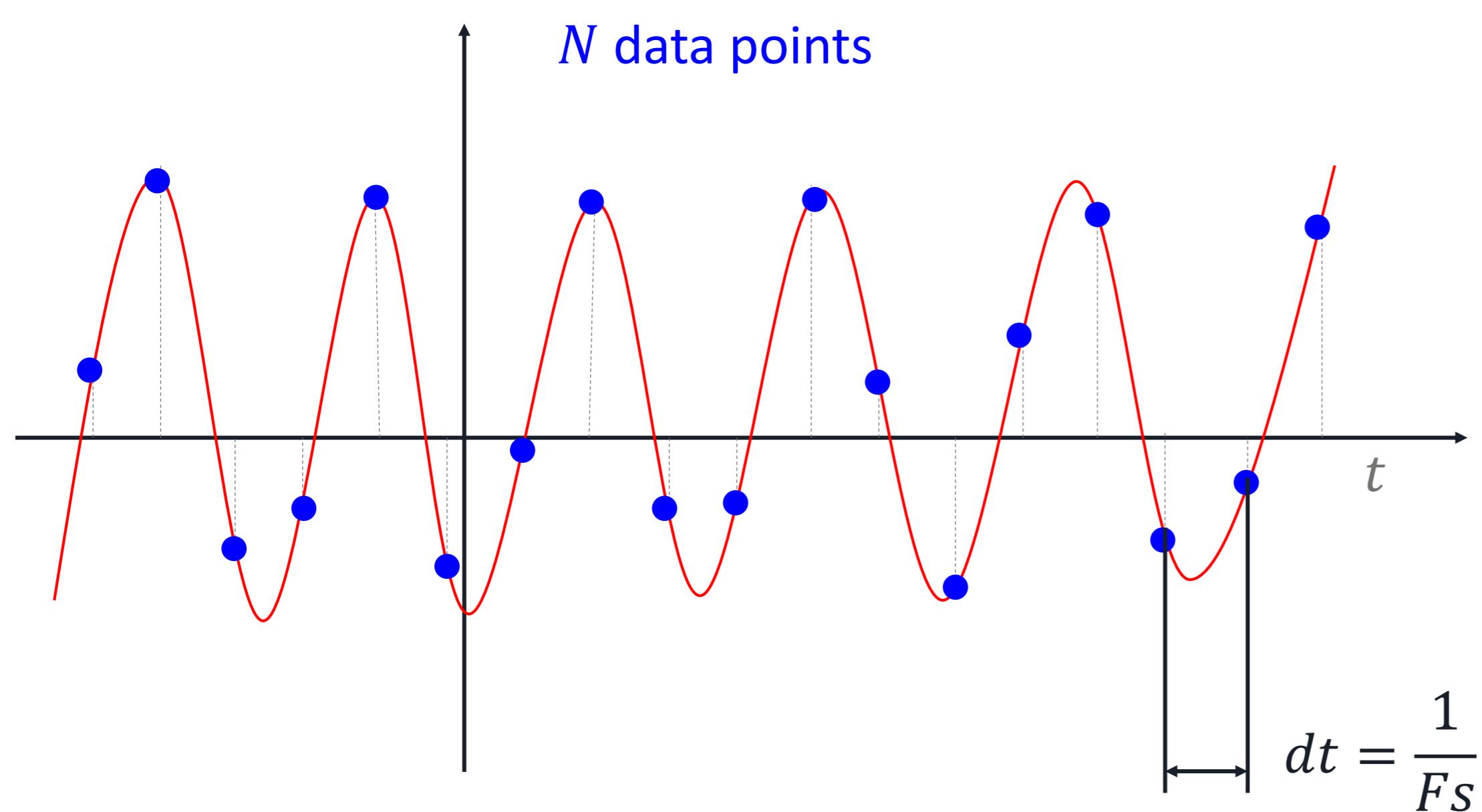
UNIVERSIDAD
DE GRANADA

https://www.youtube.com/watch?v=8_bkBL8e40Q

Aliasing

Teorema del muestreo

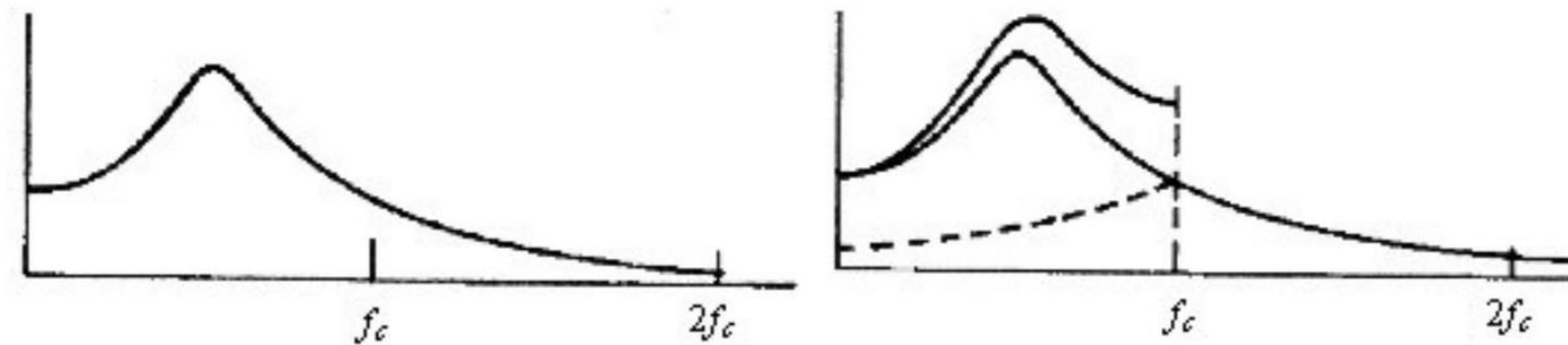
El teorema de muestreo establece que una señal se puede reproducir exactamente si se muestrea a una frecuencia F_s , donde F_s es mayor que el doble de la frecuencia máxima en la señal: **la frecuencia de Nyquist**.



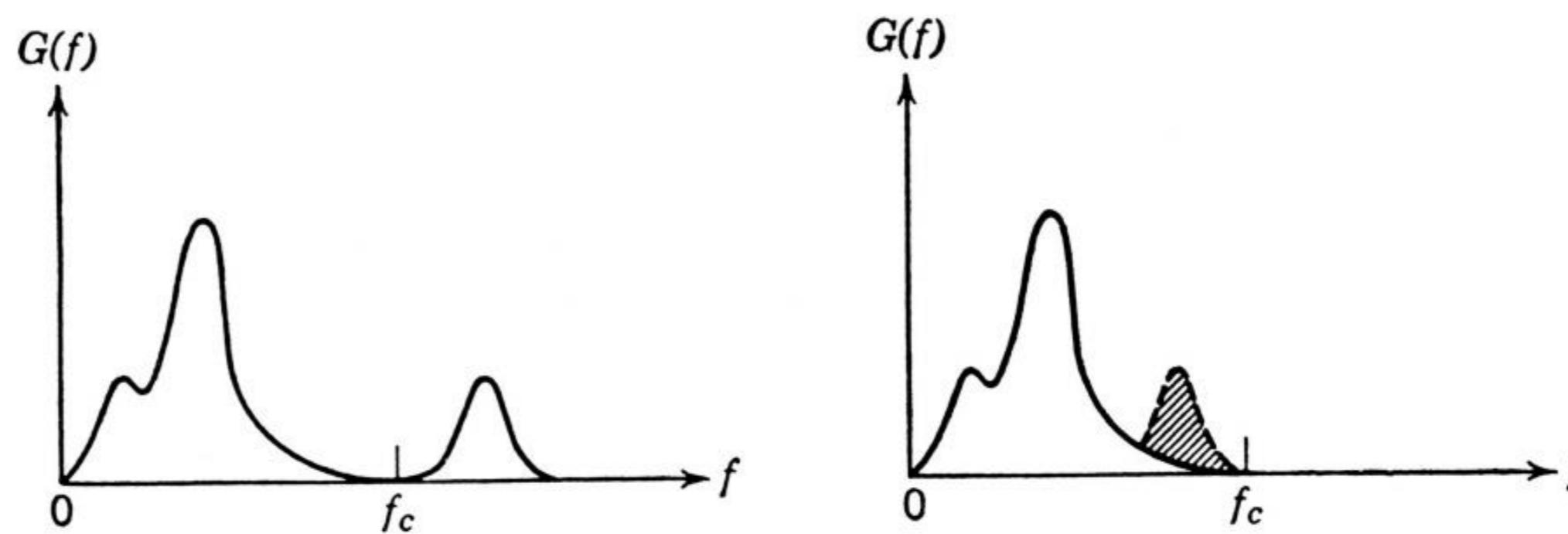
Aliasing

Teorema del muestreo

El teorema de muestreo establece que una señal se puede reproducir exactamente si se muestrea a una frecuencia F_s , donde F_s es mayor que el doble de la frecuencia máxima en la señal: **la frecuencia de Nyquist**.



Distorsión producida por el *aliasing* en una señal con frecuencias superiores a la de Nyquist.

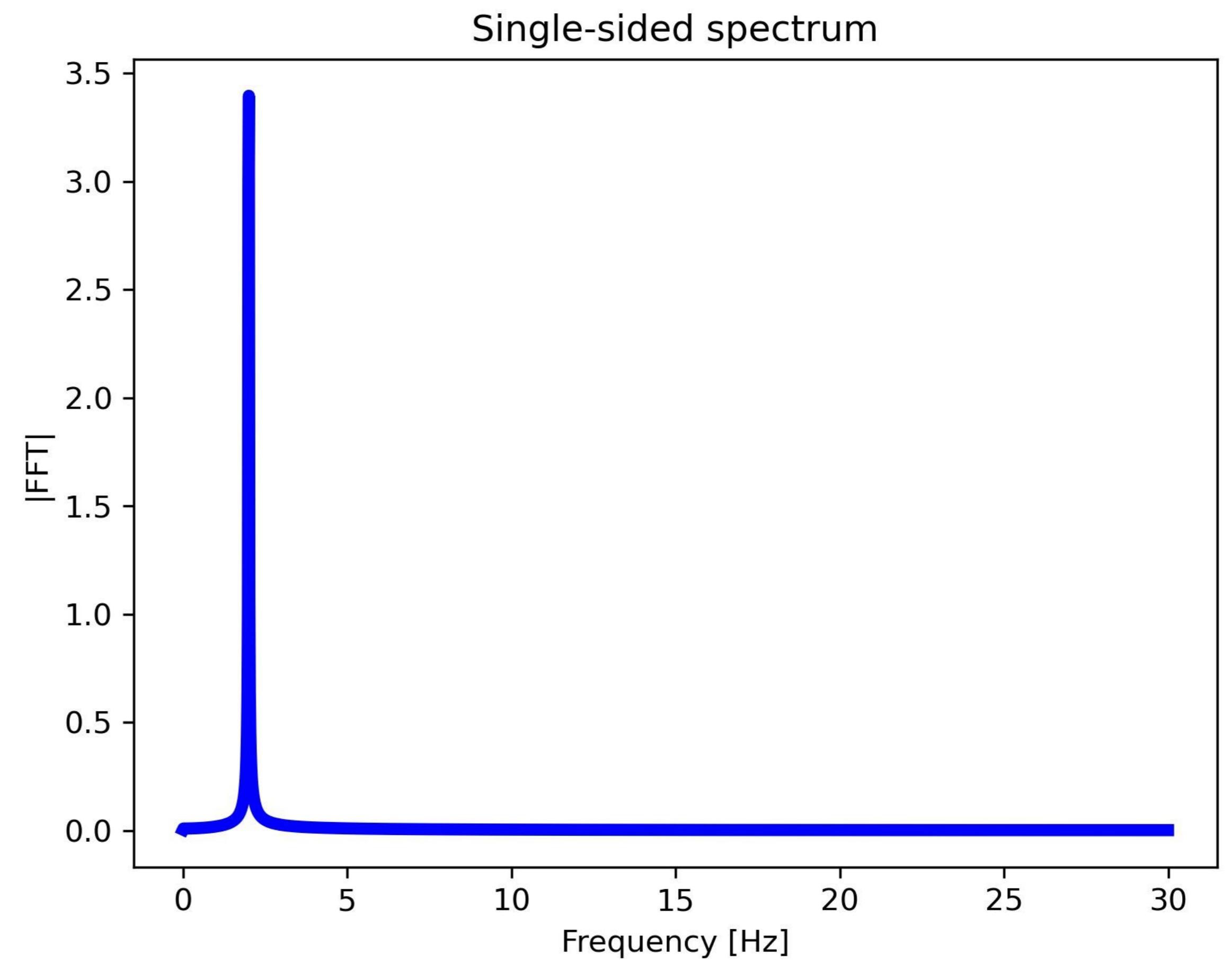
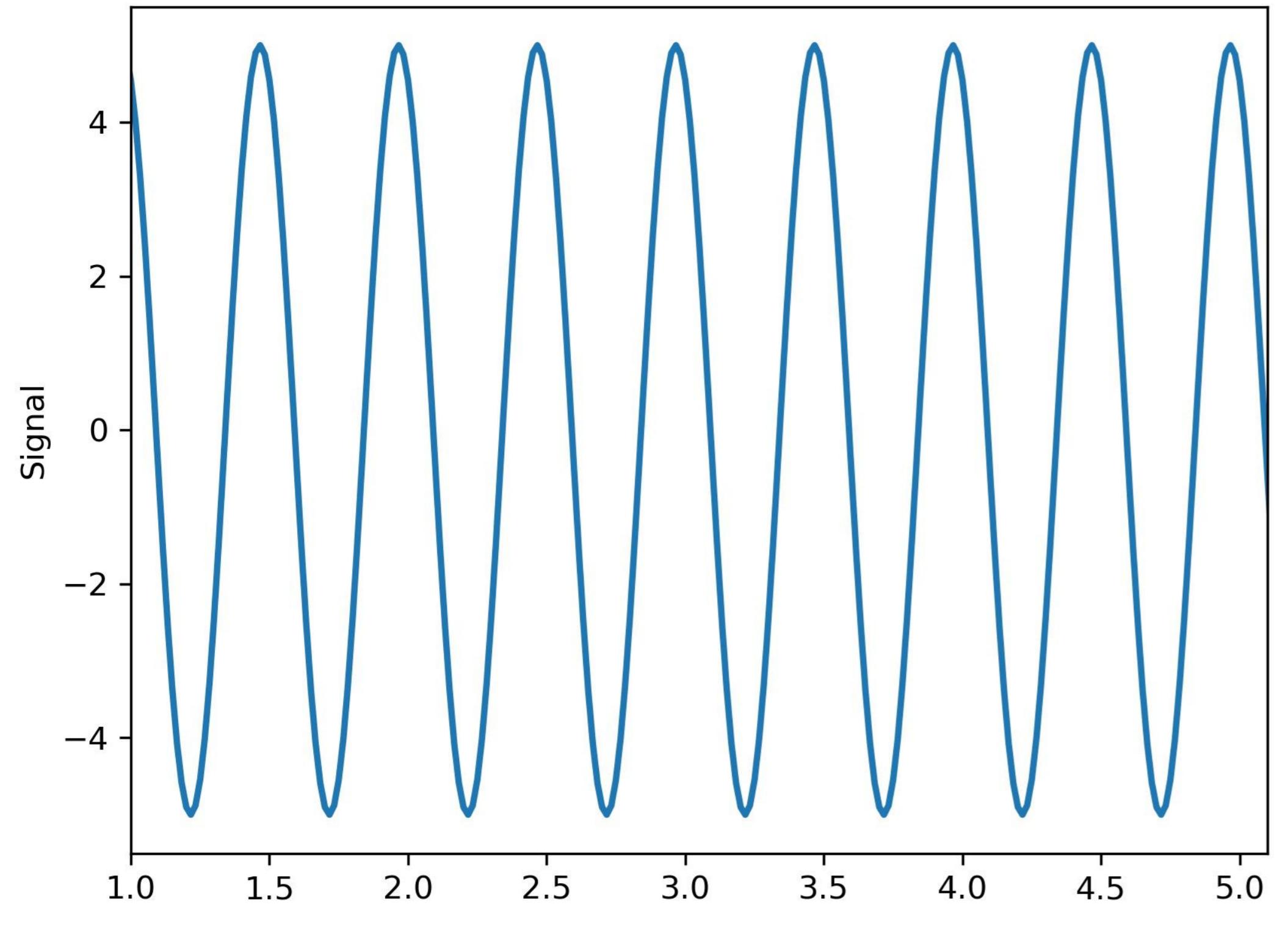


Comparación entre un autoespectro real y el mismo con *aliasing*.



Frequency Leakage.

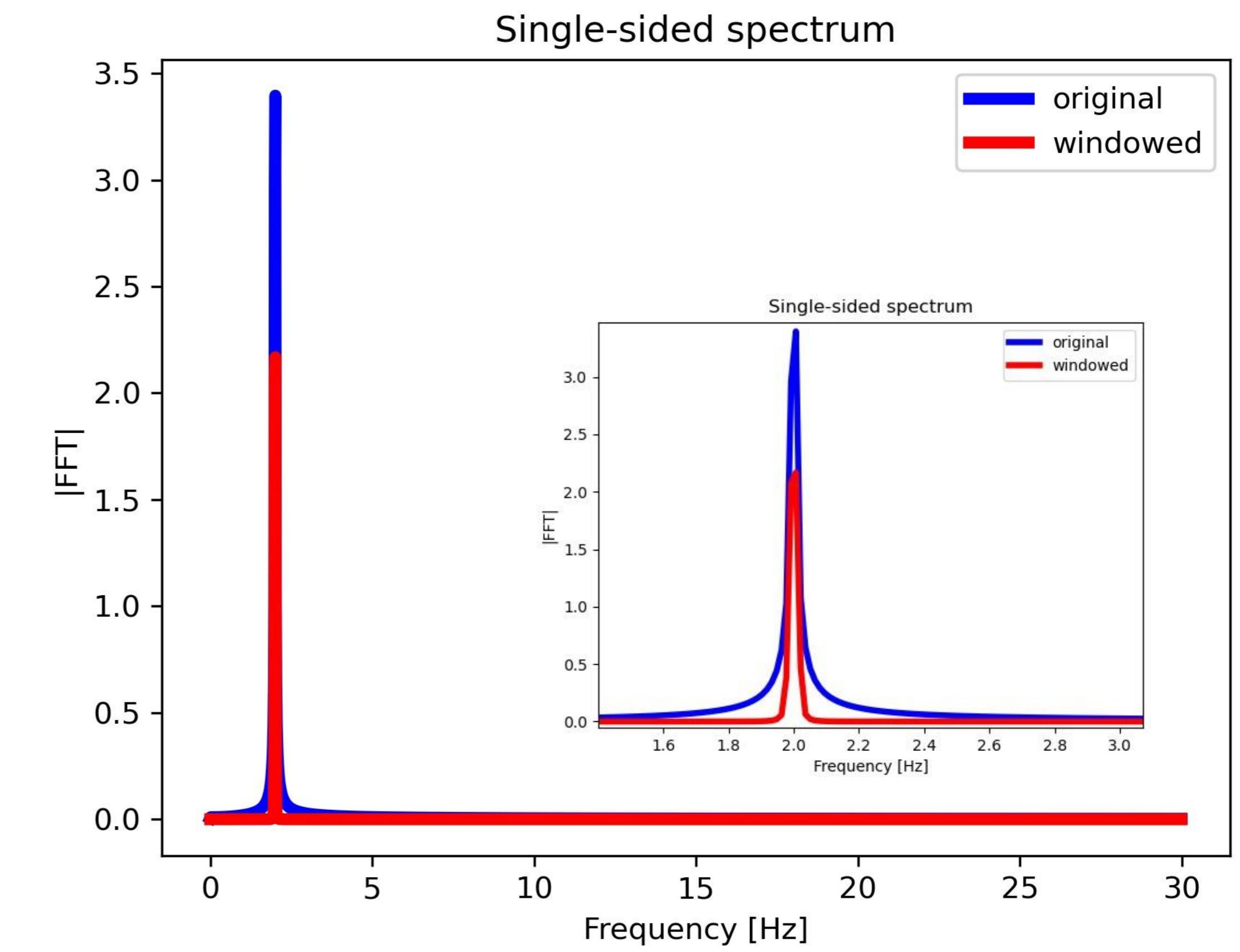
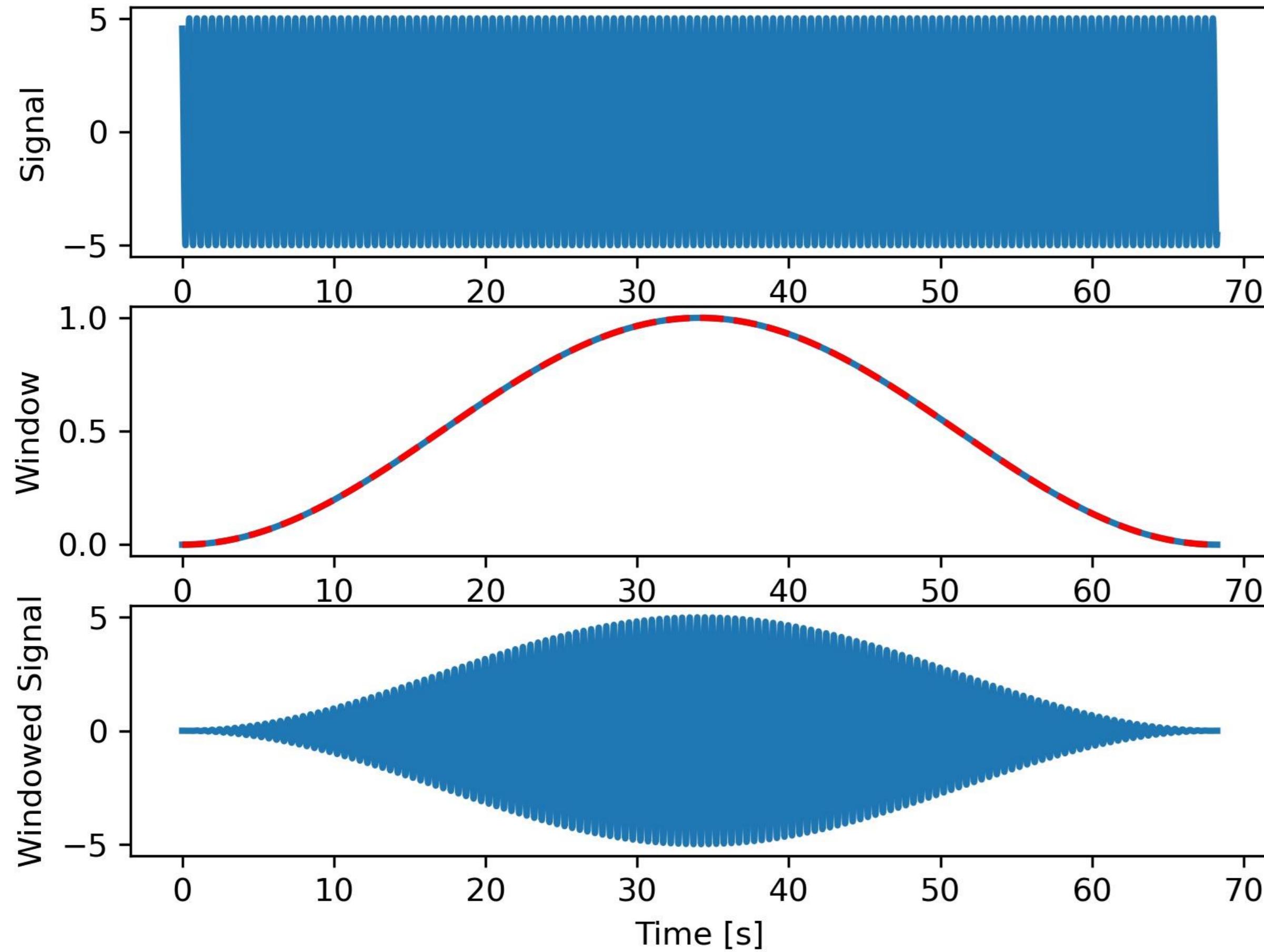
Frequency Leakage



Frequency Leakage

Hann (Hanning) function

$$Hann[n] = \sin^2\left(\frac{\pi n}{N}\right)$$





Cross-correlation.

Cross-correlation

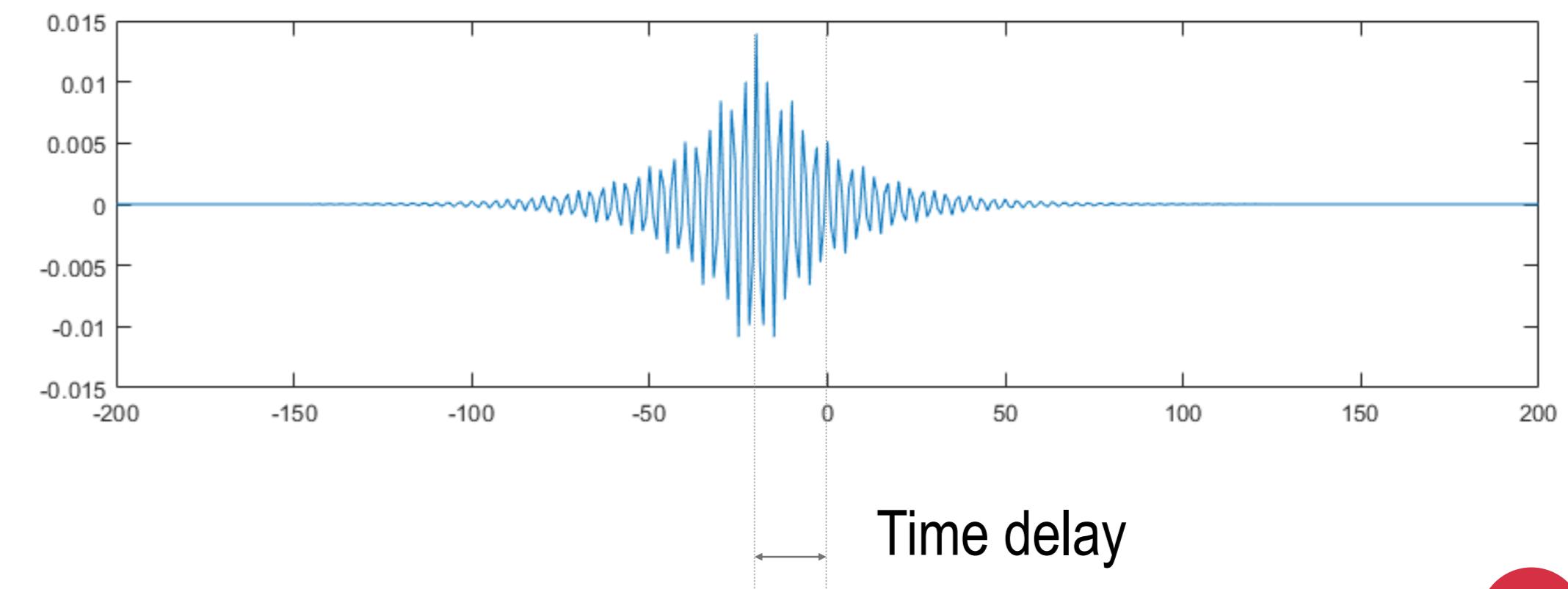
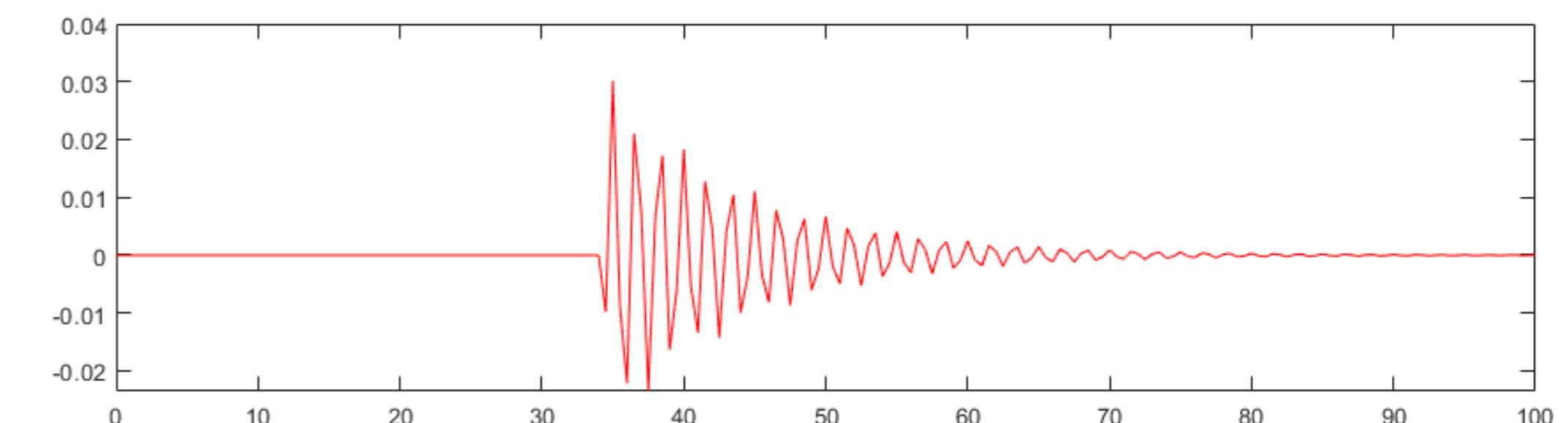
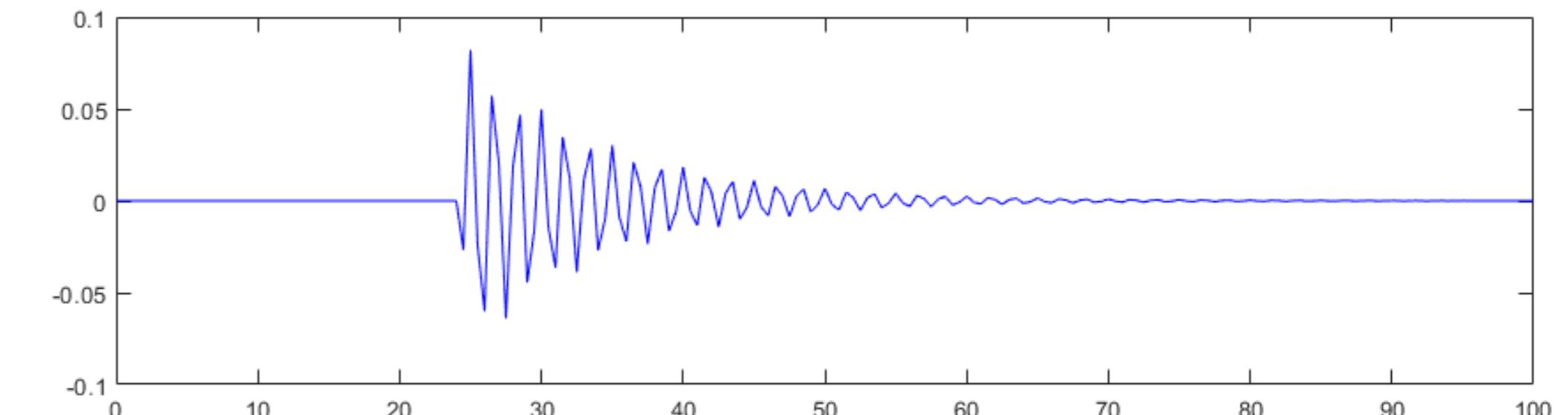
Medida de similitud de dos series en función del desplazamiento de una con respecto a la otra.

$$r_{fg}(\tau) = \int_{-\infty}^{\infty} \overline{f(t)}g(t + \tau)dt$$

Forma discreta:

$$r_{fg}(n) = \sum_{m=-\infty}^{\infty} \overline{f(m)}g(m + n)$$

$f = g$ Auto-correlation



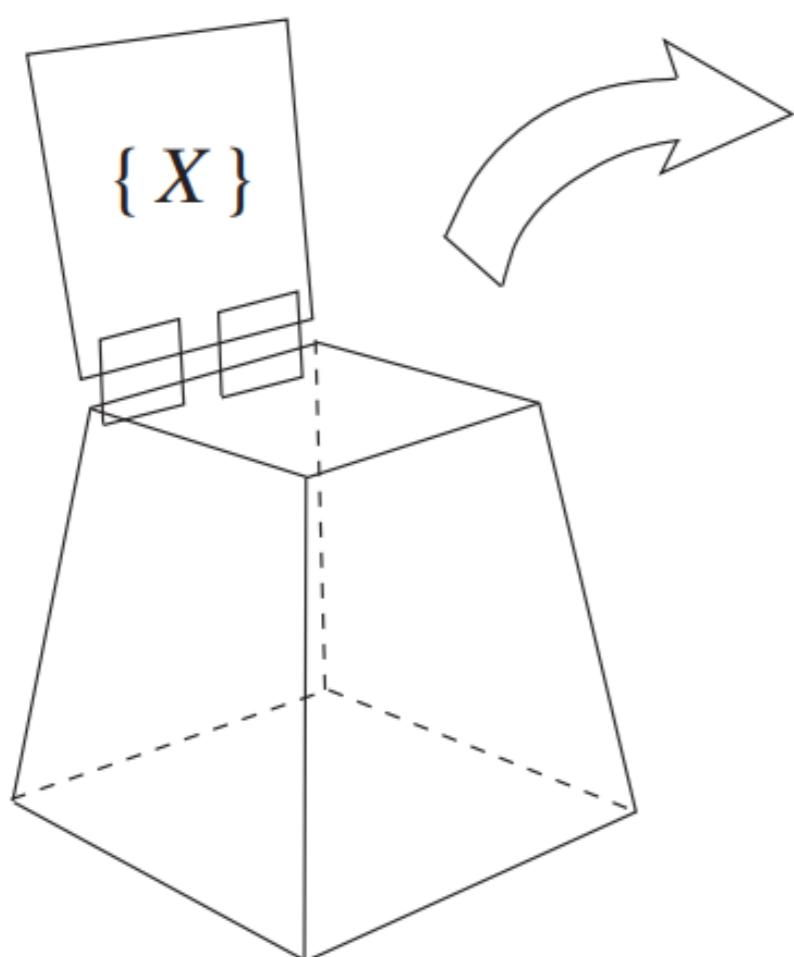


UNIVERSIDAD
DE GRANADA

Random Signals

Random signals

En la mayoría de las aplicaciones prácticas, tenemos que tratar con señales que no pueden describirse en función del tiempo, ya que sus formas de onda siguen un patrón aleatorio. Tomemos, por ejemplo, el ruido eléctrico inherente a cualquier sensor. A pesar de la falta de un conocimiento exacto de los valores de la señal, es posible analizar y extraer información de la señal empleando las herramientas matemáticas disponibles para tratar con señales aleatorias.



Definición (variable aleatoria discreta)

Una serie aleatoria discreta $X = \{a_0, \dots, a_n, \dots\}$ se caracteriza por una función de distribución de probabilidad (probability density function pdf) como:

$$p_X(n) = \Pr(X = a_n)$$

La probabilidad de que X pertenezca al intervalo $[a, b]$ viene dada por:

$$\Pr(X \in [a, b]) = \sum_{n \geq 0, a_n \in [a, b]} p_X(n)$$

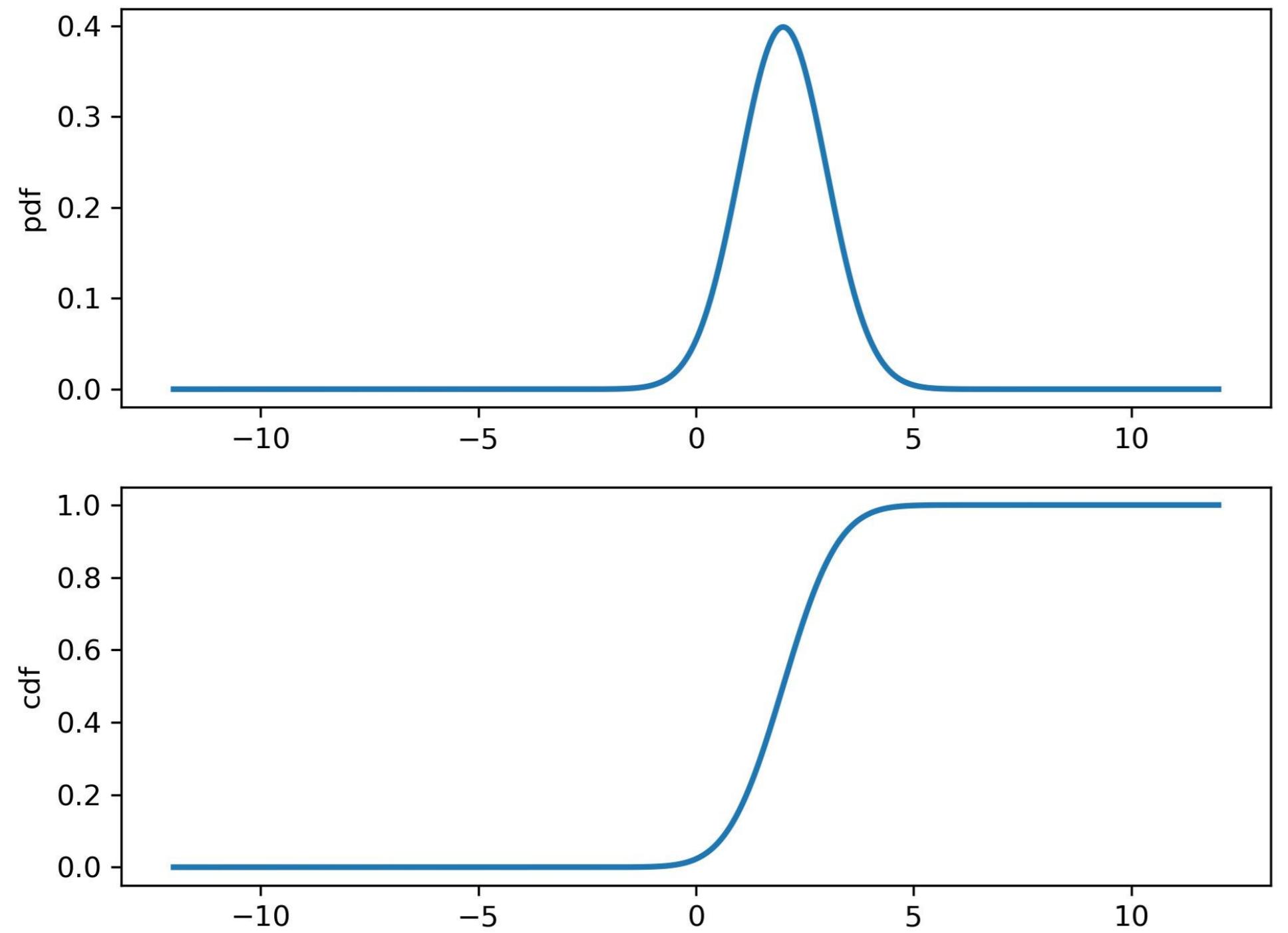
Esto conduce a la definición de la función de distribución acumulativa (cumulative distribution function cdf)

$$F_X(x) = \Pr(X \leq x) = \sum_{n \geq 0, a_n \leq x} p_X(n)$$

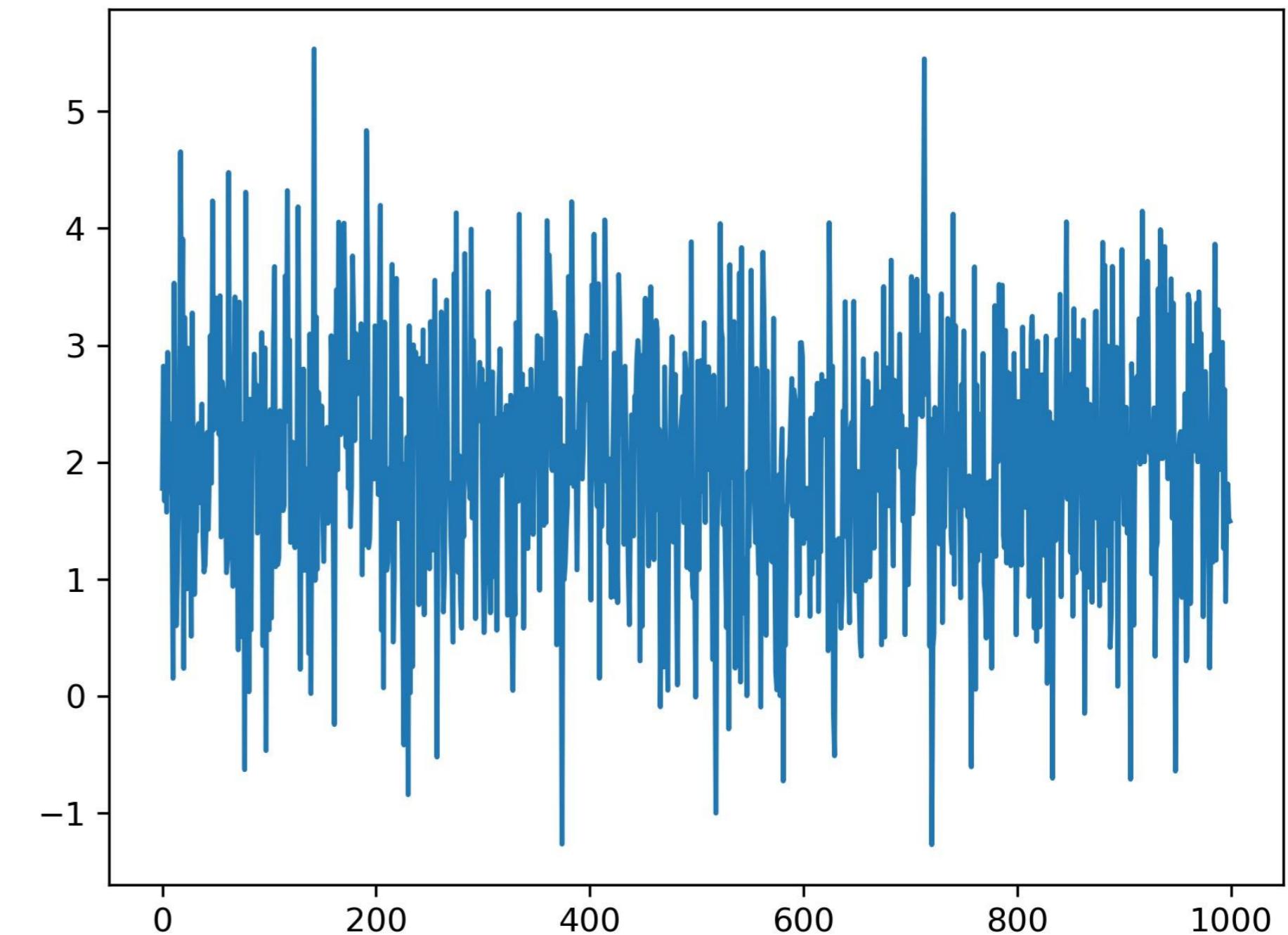
Random signals

Eg. Gaussian signal

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



Sampling



Random signals

Eg. Gaussian signal

Statistical moments

Mean	$\mu = E(X)$
Variance	$\sigma^2 = E(X - \mu)^2$
Skewness	$S = \frac{E(X - \mu)^3}{\sigma^3}$
Kurtosis	$k = \frac{E(X - \mu)^4}{\sigma^4}$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$$\begin{cases} \mu \\ \sigma^2 \\ S = \frac{E(X - \mu)^3}{\sigma^3} = 0 \\ k = \frac{E(X - \mu)^4}{\sigma^4} = 3 \end{cases}$$



UNIVERSIDAD
DE GRANADA

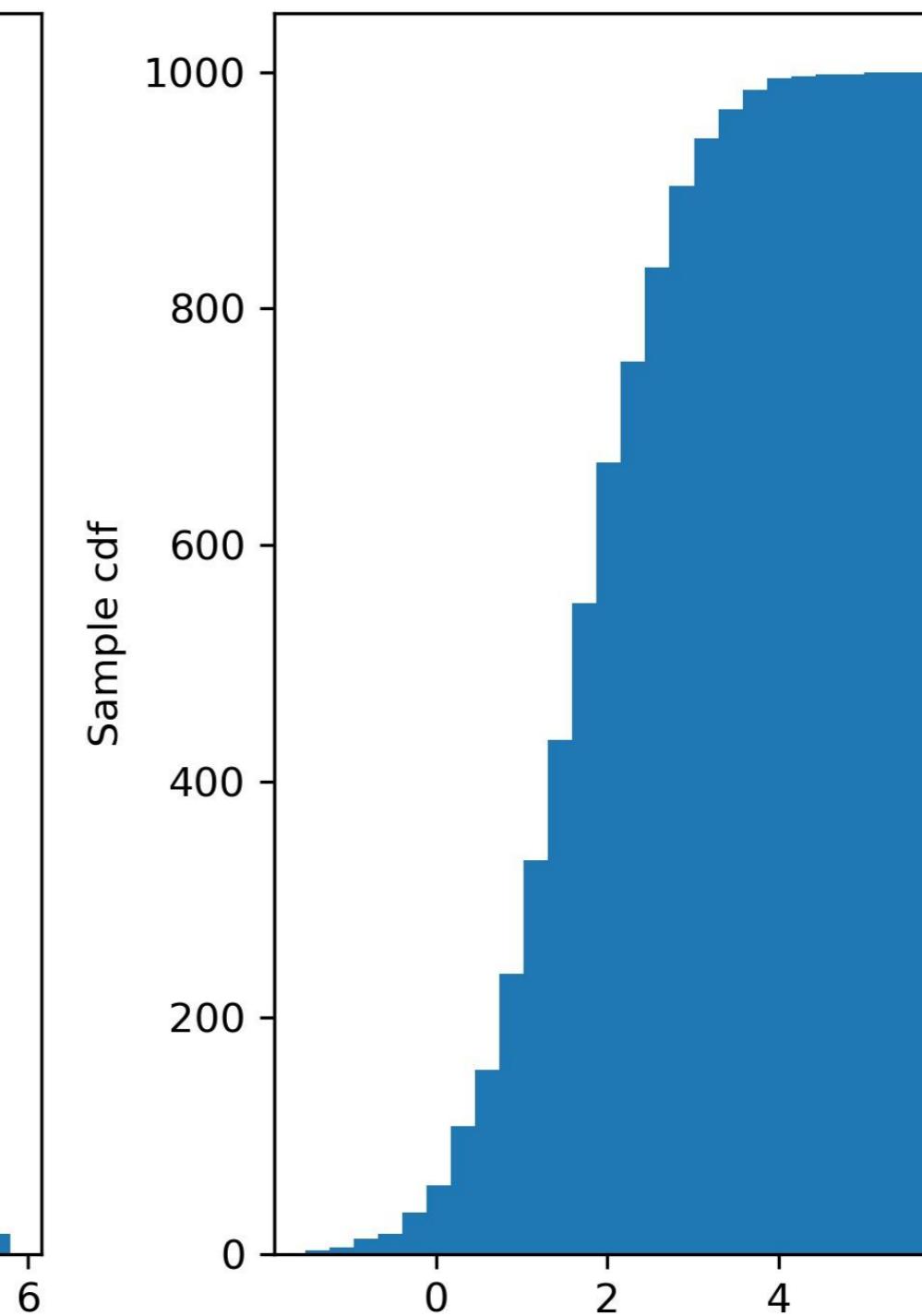
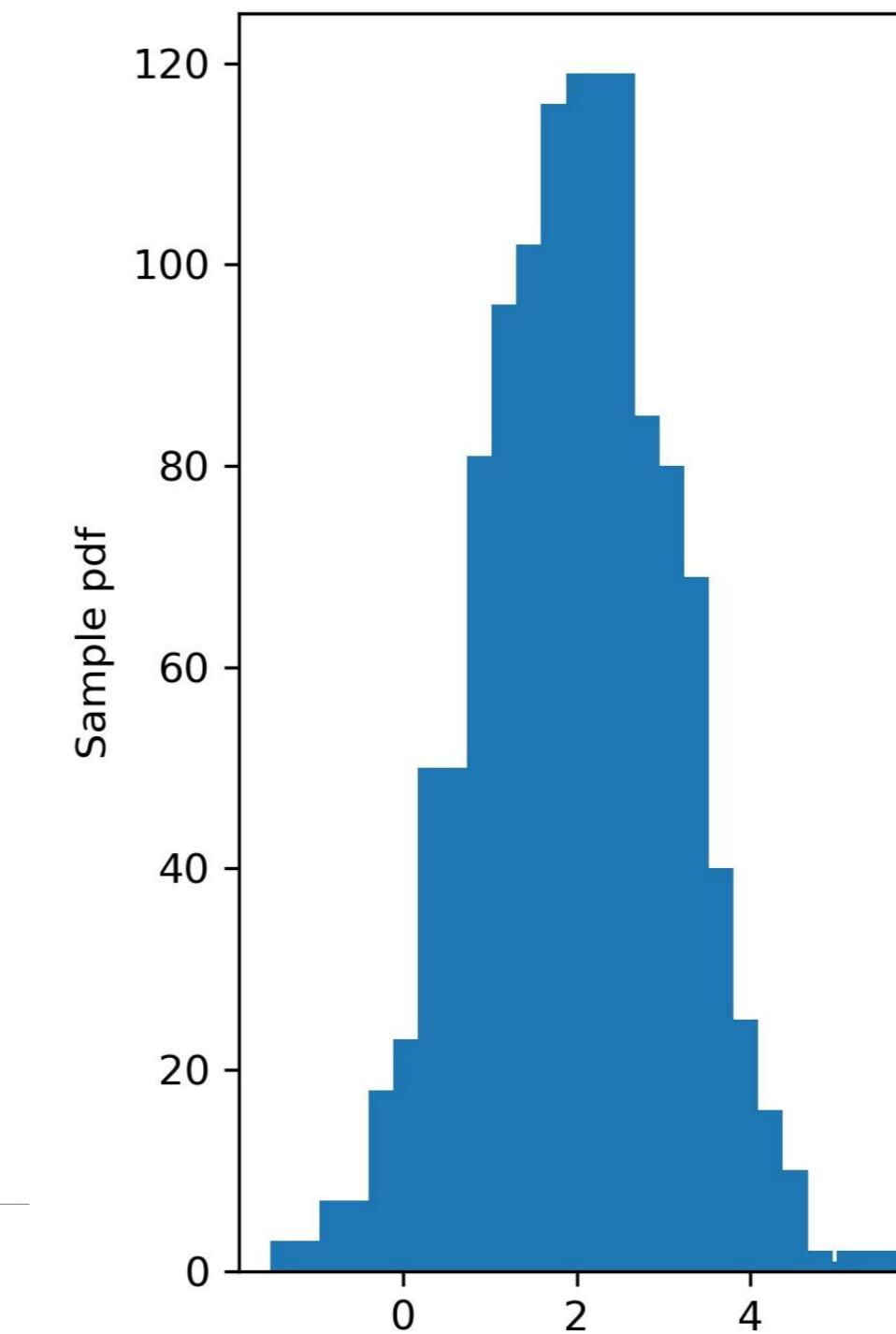
[I_Gaussian_random_signal.py](#)

```
# Mean value
mean_X = (1/N)*np.sum(X)
np.mean(X)

# Variance
var = (1/(N-1))*np.sum((X-mean_X)**2);
stdvar = np.sqrt(var)
np.std(X, ddof=1)

# Skewness
skewnessval = (1/(N))*np.sum((X-mean_X)**3)/(np.sqrt((1/(N))*np.sum((X-
mean_X)**2)))**3
stats.skew(X)

# Kurtosis
Kurtosis = (1/(N))*np.sum((X-mean_X)**4)/(np.sqrt((1/(N))*np.sum((X-
mean_X)**2)))**4
stats.kurtosis(X, fisher = False)
```

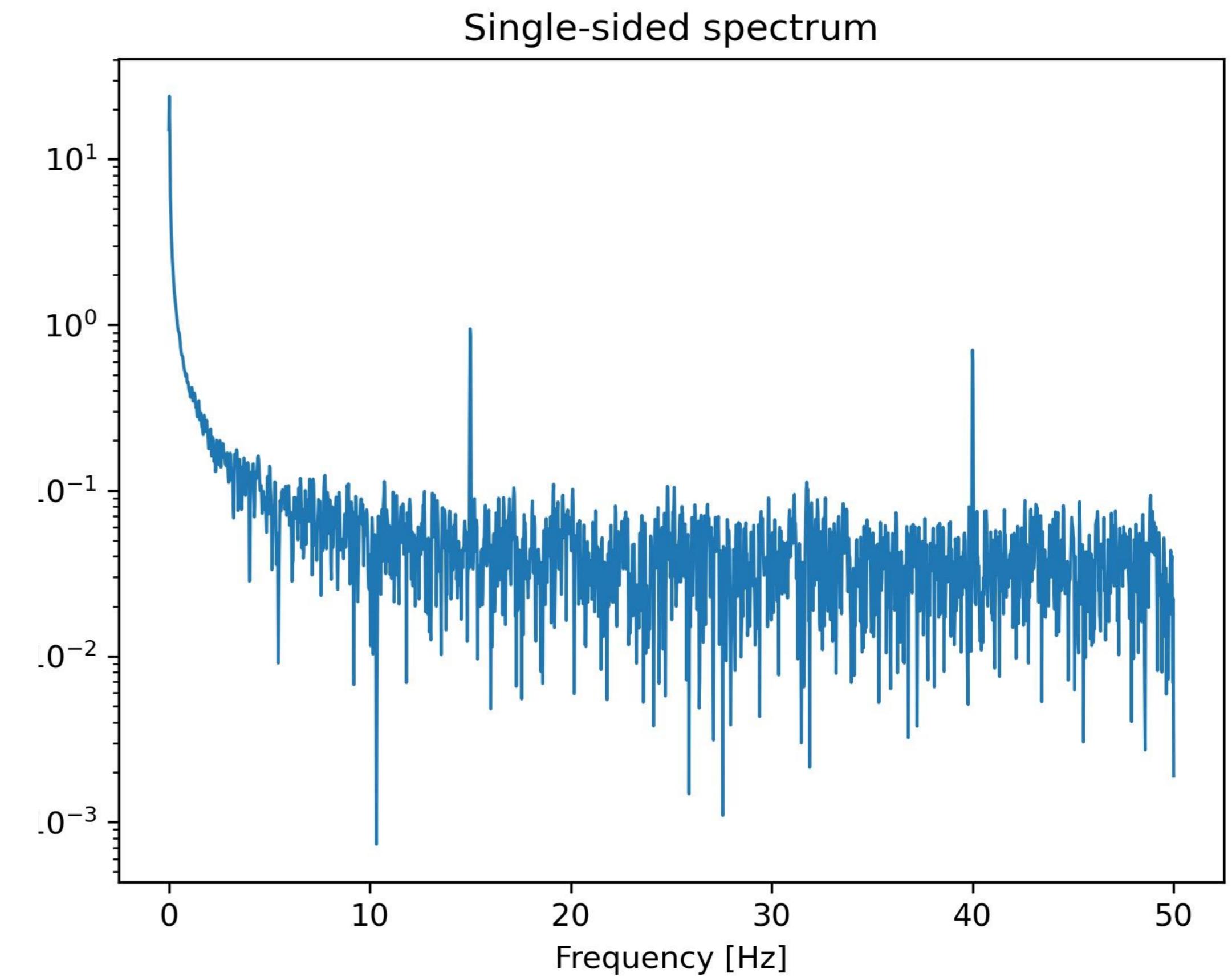
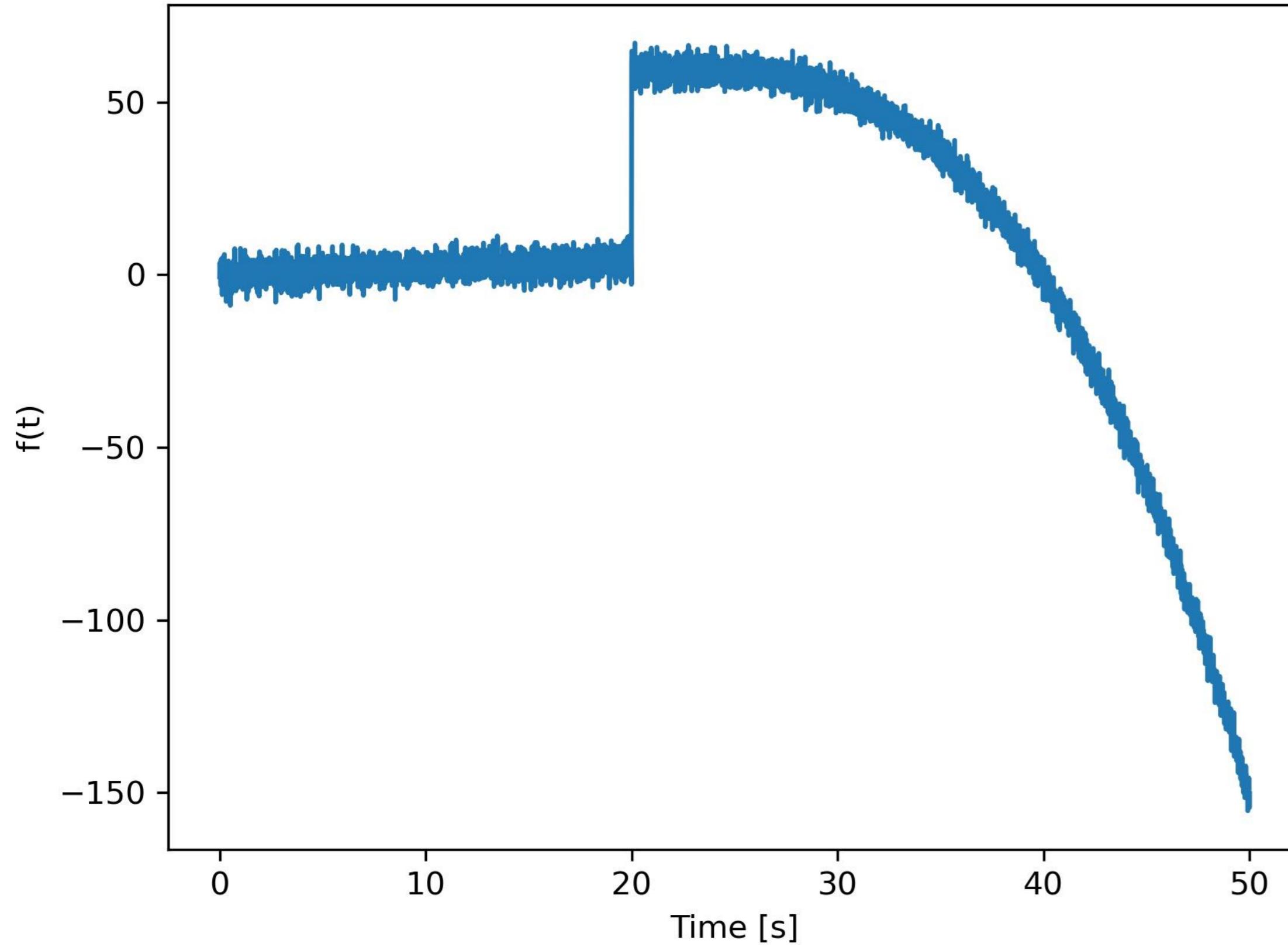


Filtering



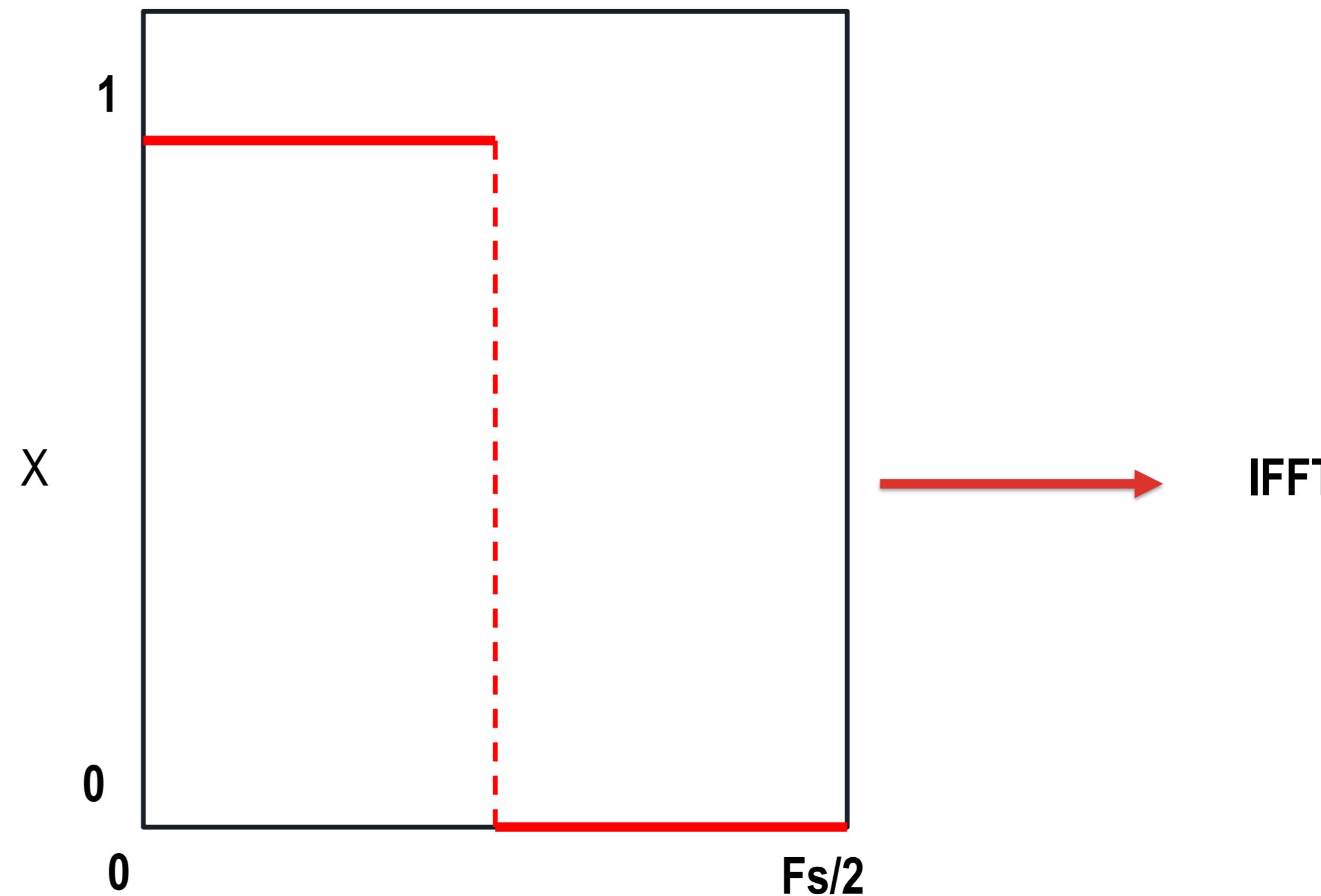
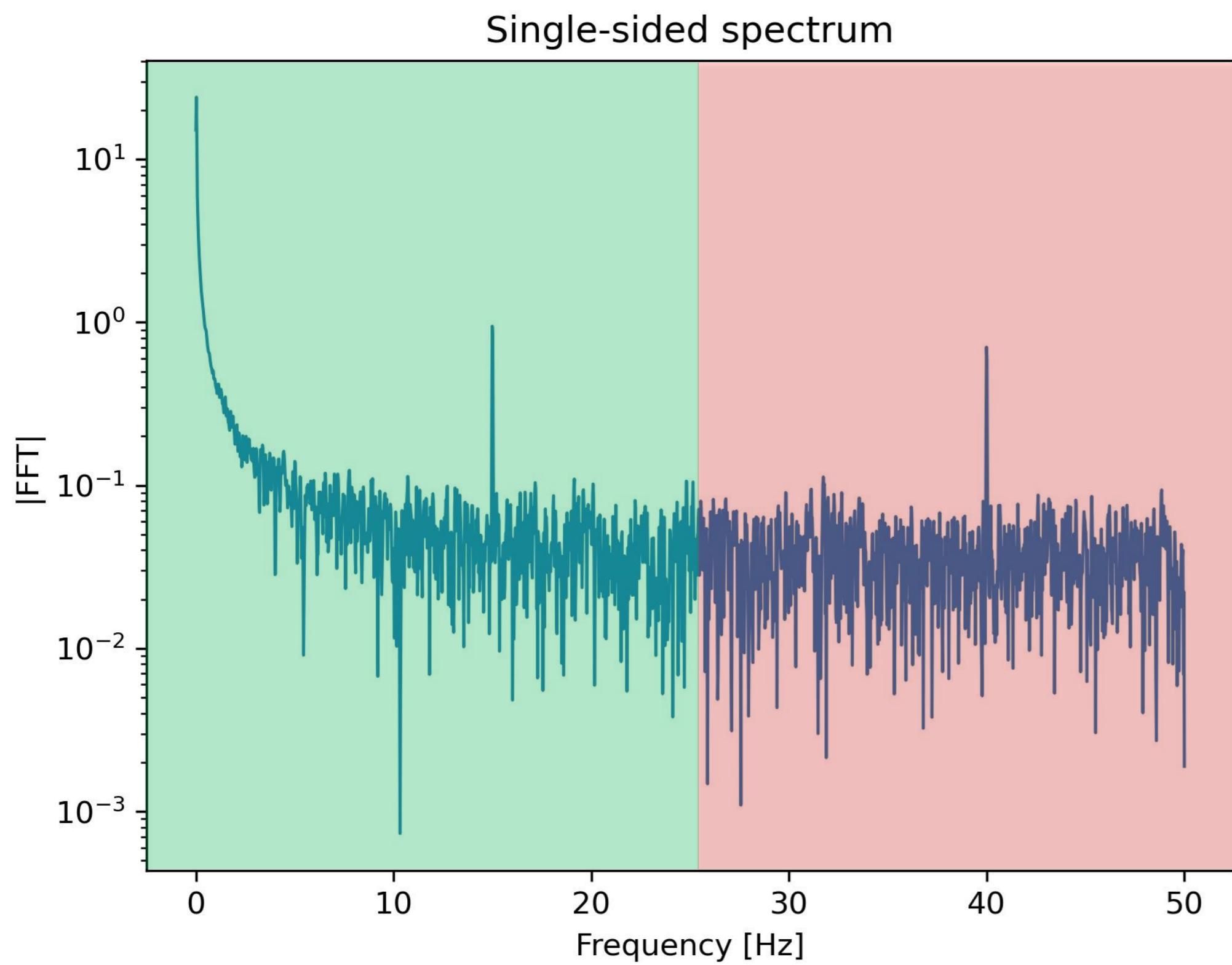
UNIVERSIDAD
DE GRANADA

Filtering



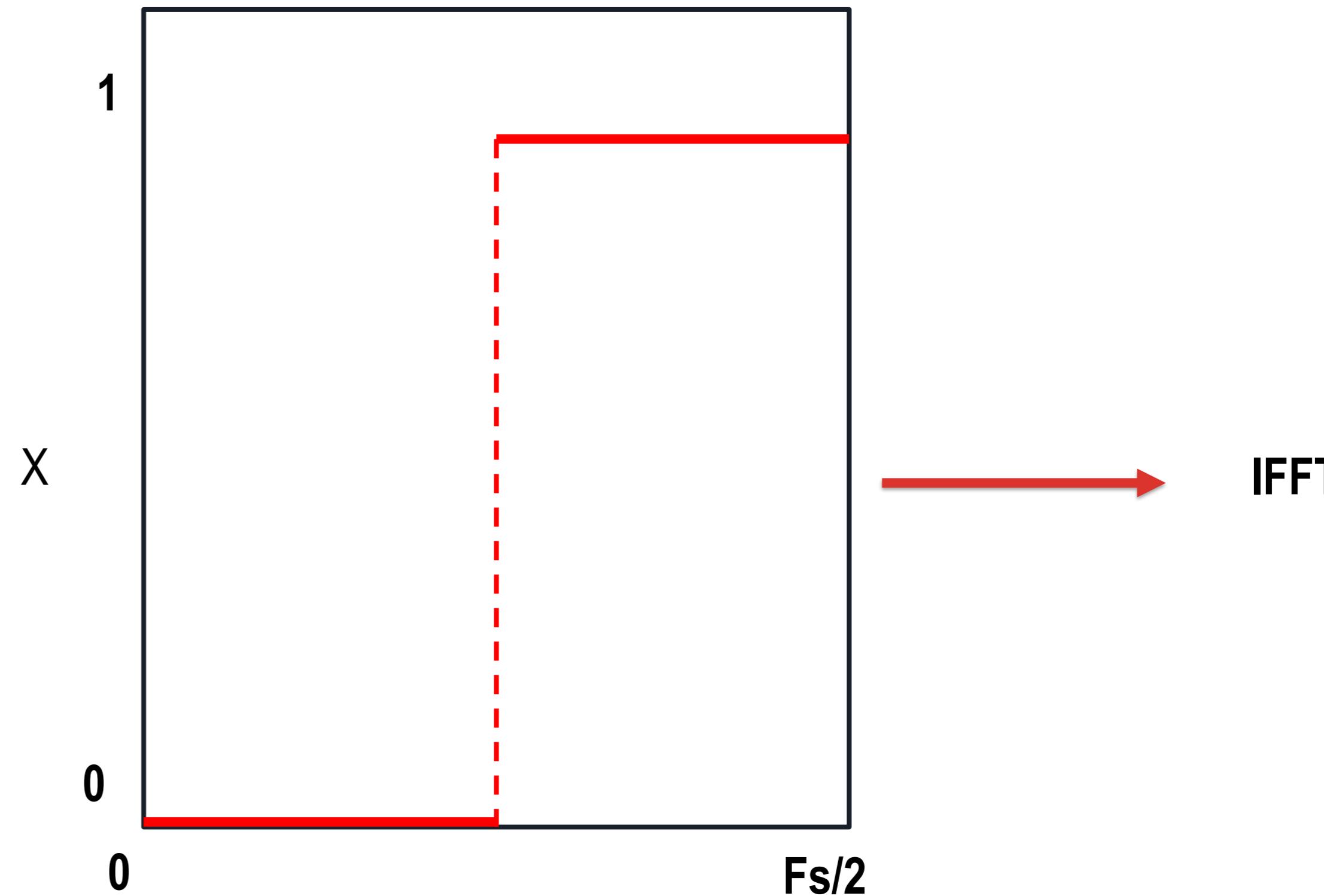
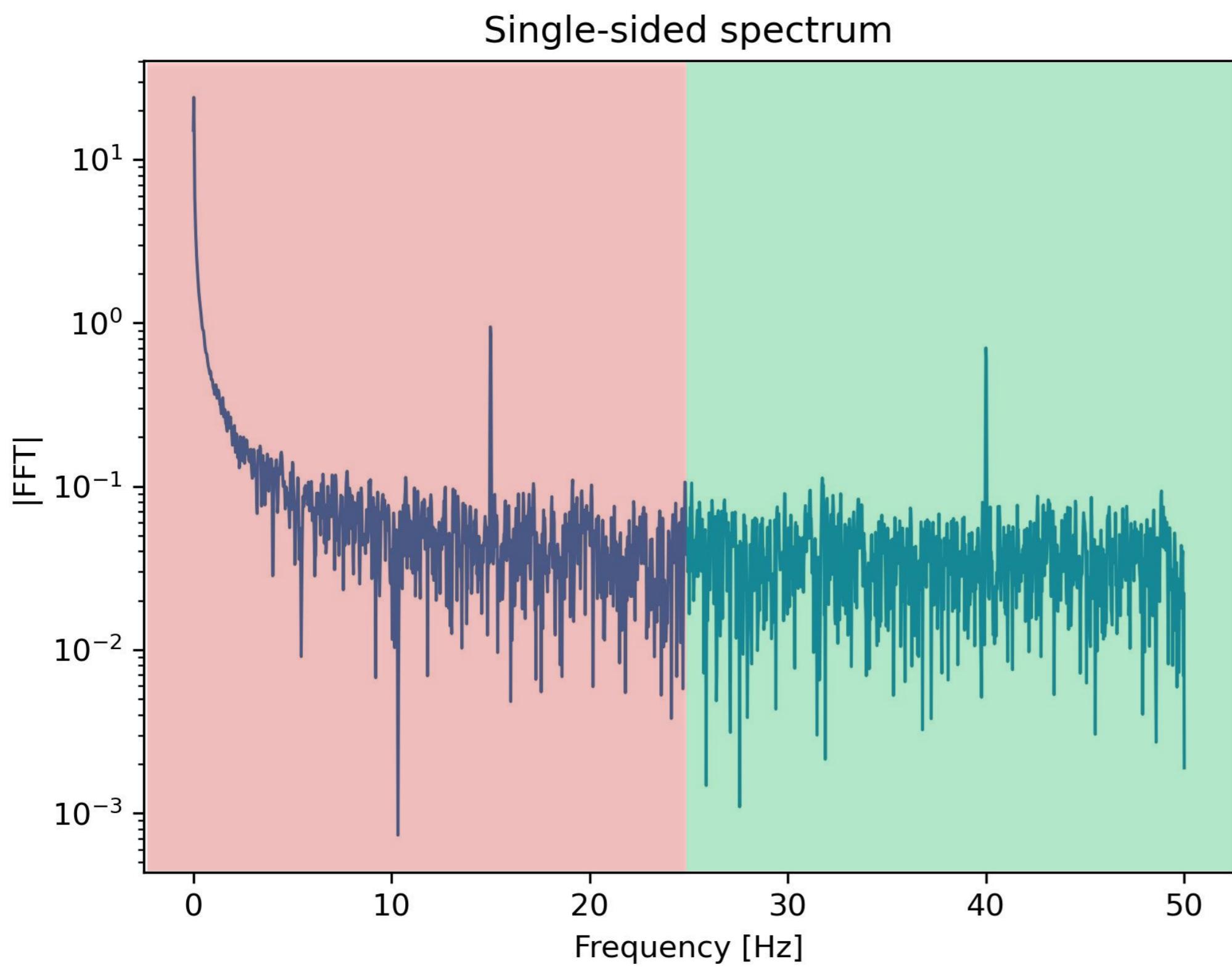
Filtering

Low-pass filter



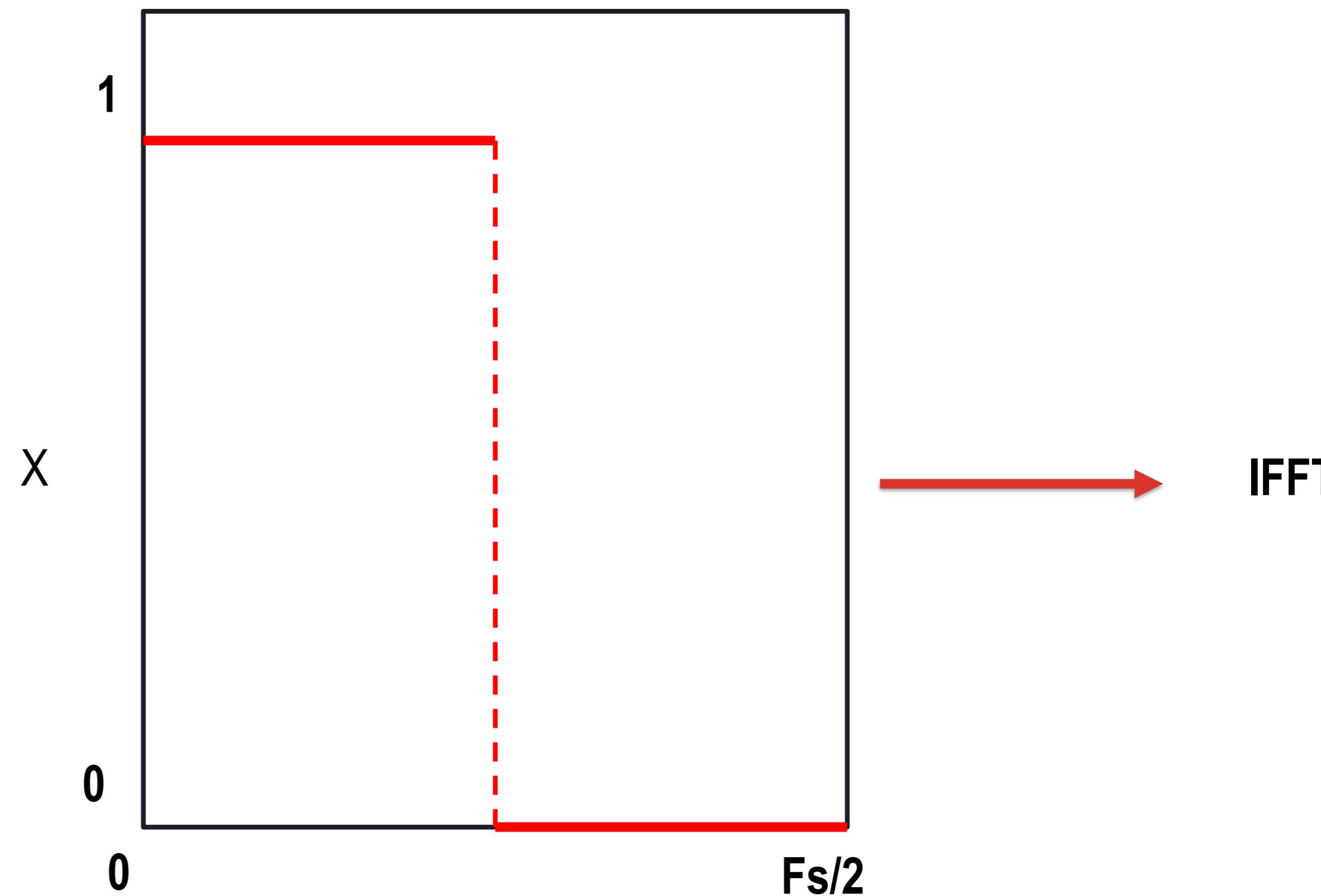
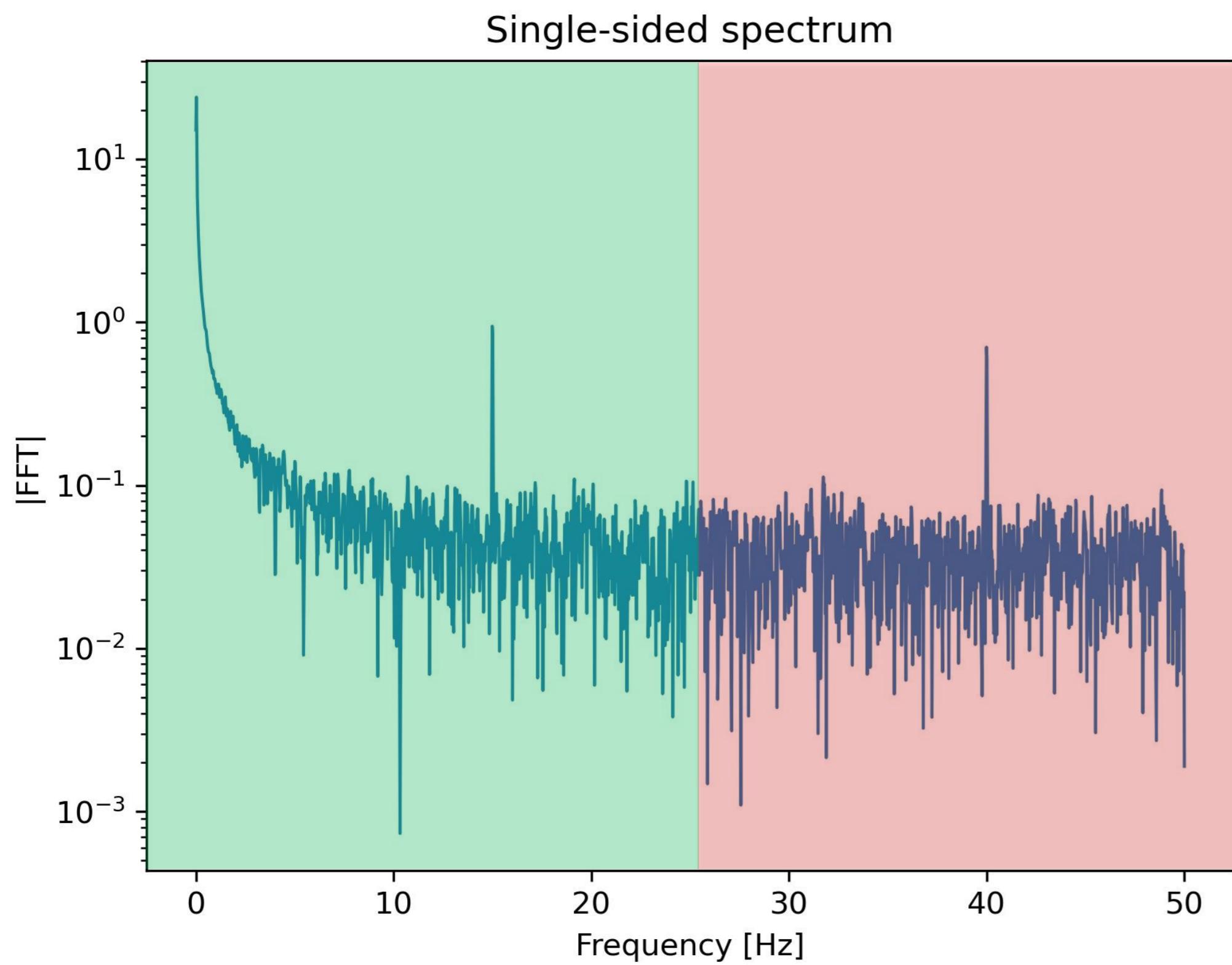
Filtering

High-pass filter

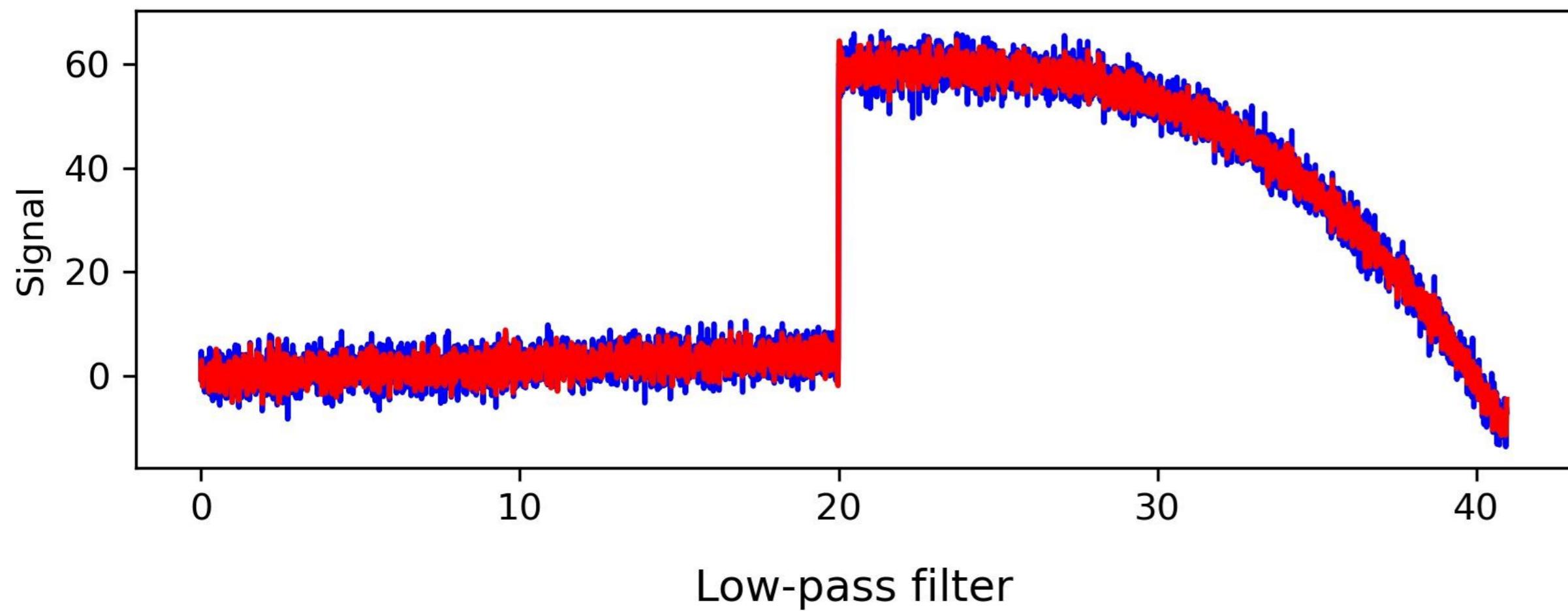


Filtering

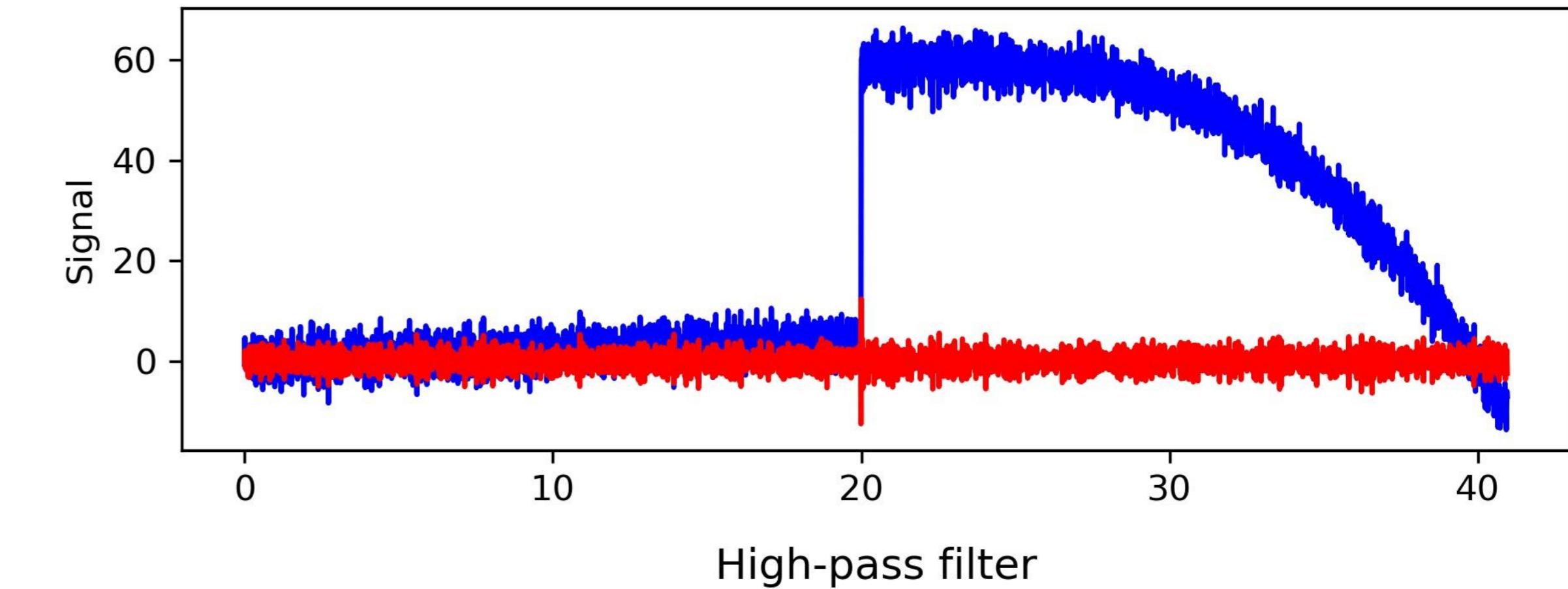
Band-pass filter



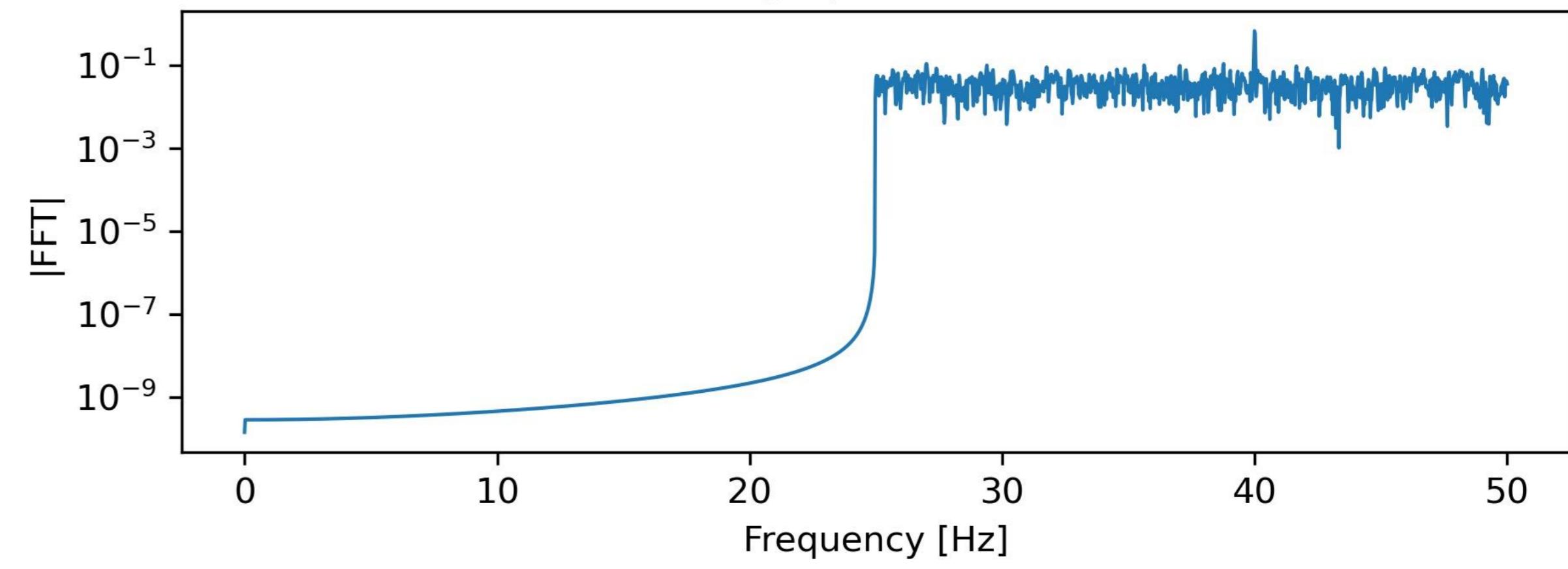
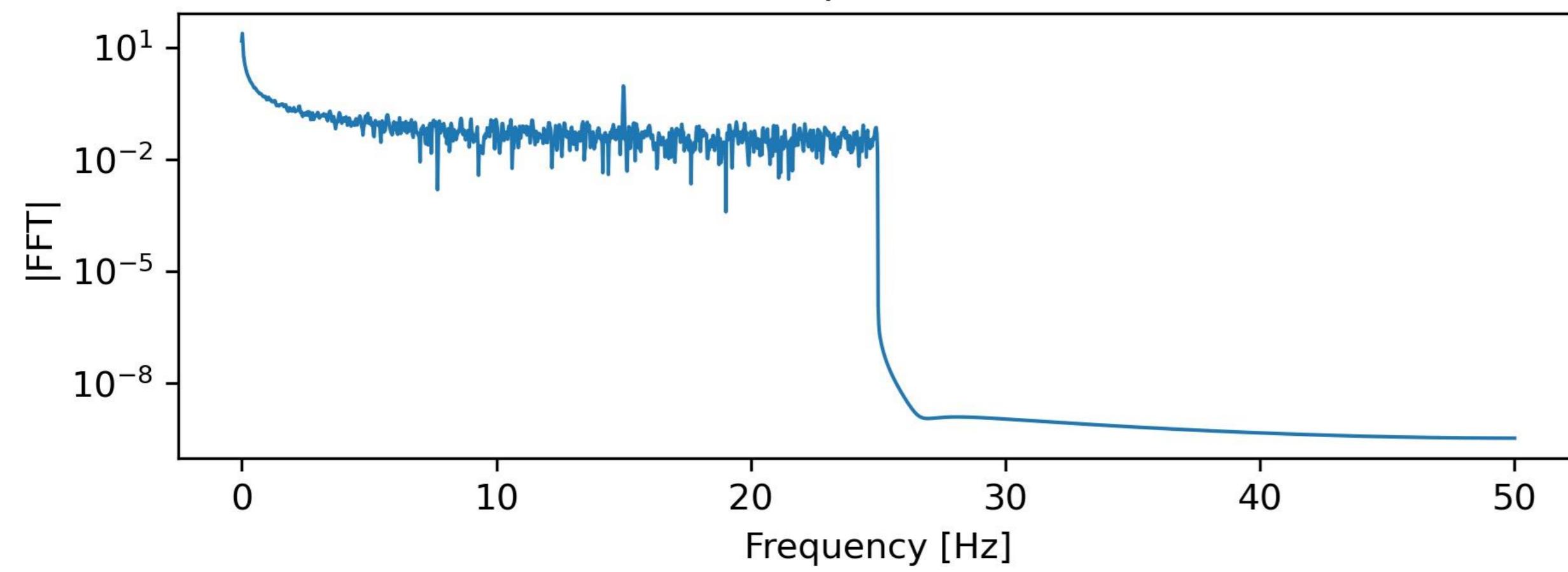
Filtering



Low-pass filter



High-pass filter

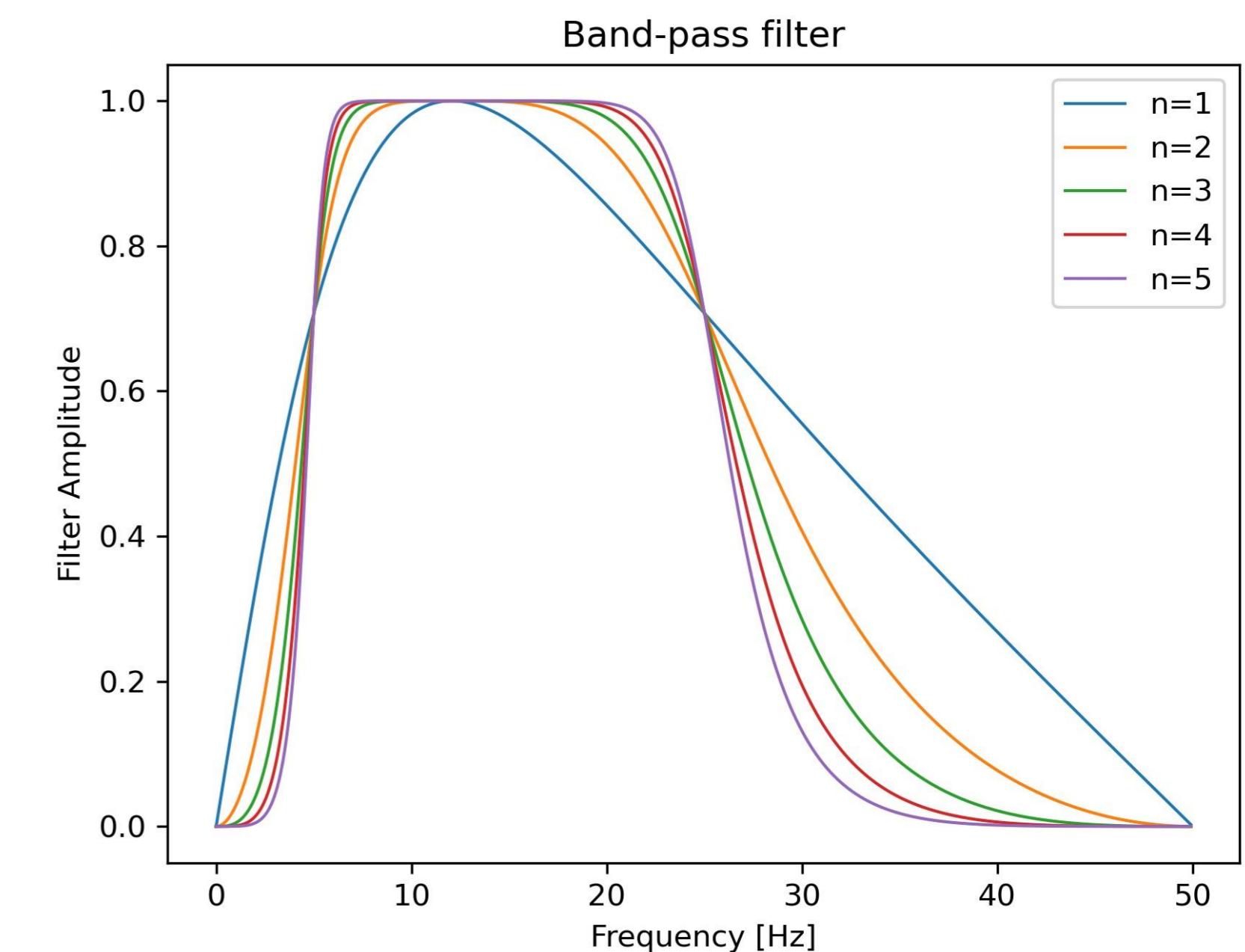
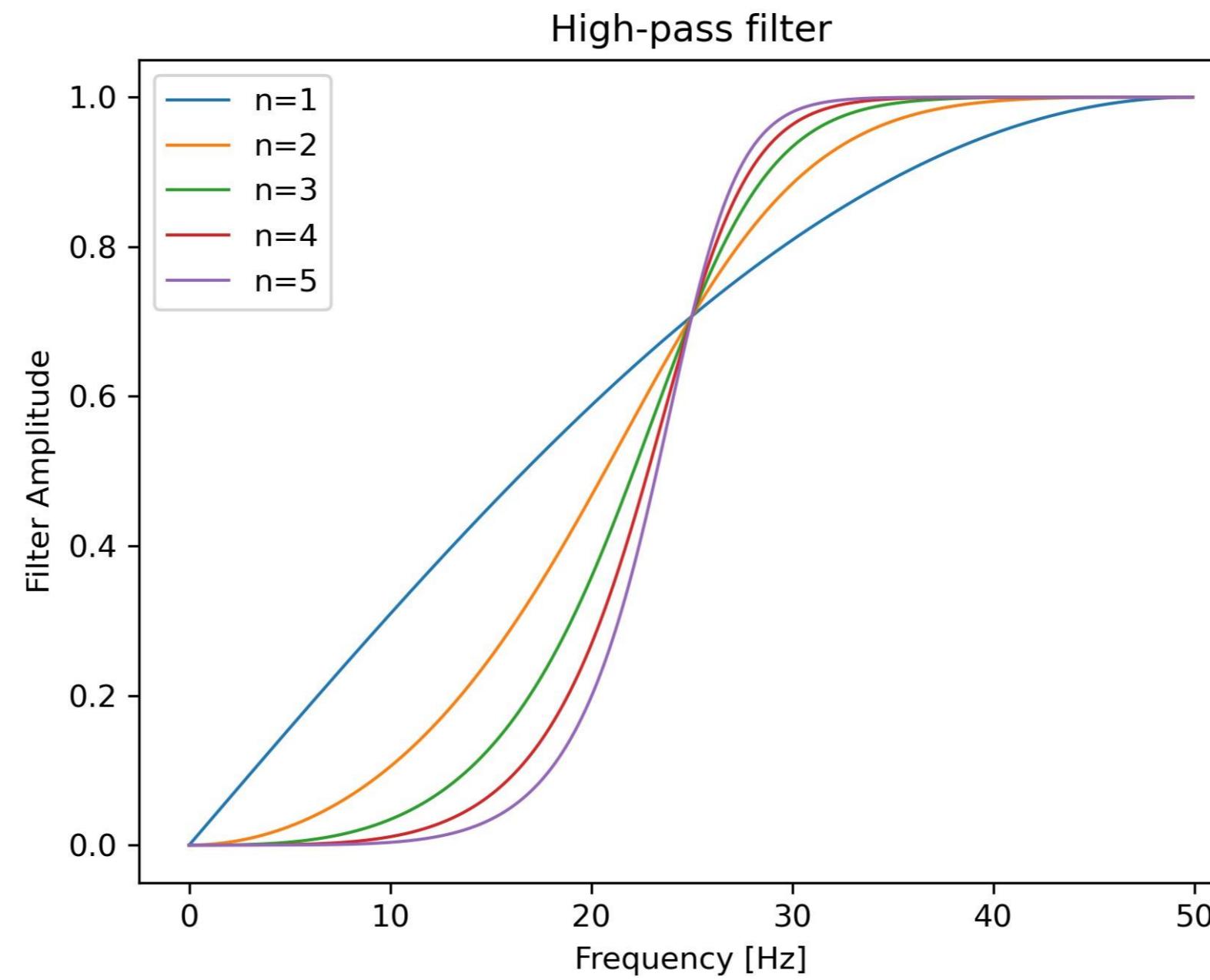
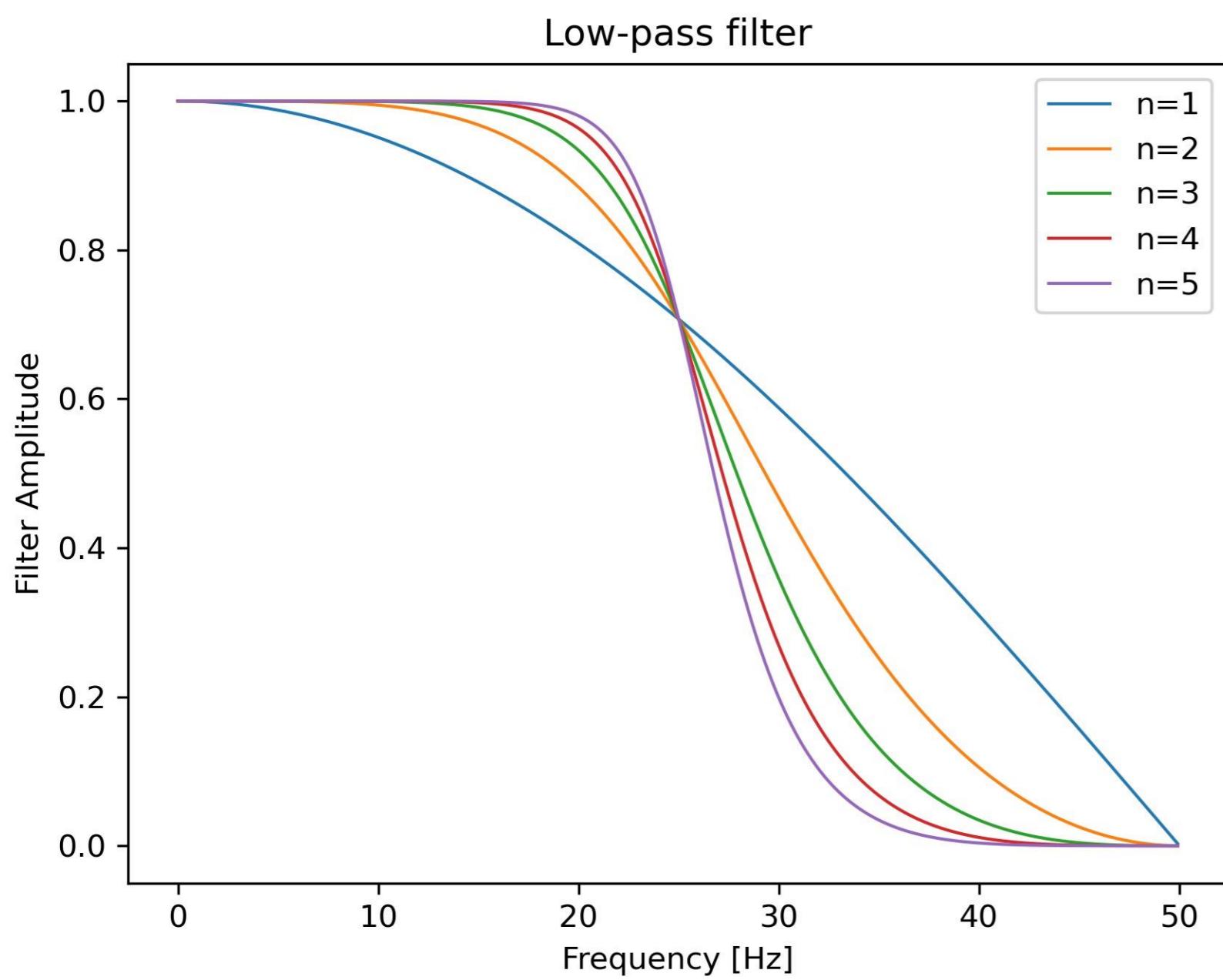


Filtering Butterworth

```
b, a = signal.butter(ij, fc/(Fs/2), 'low')  
w, histbutter = signal.freqz(b,a)
```

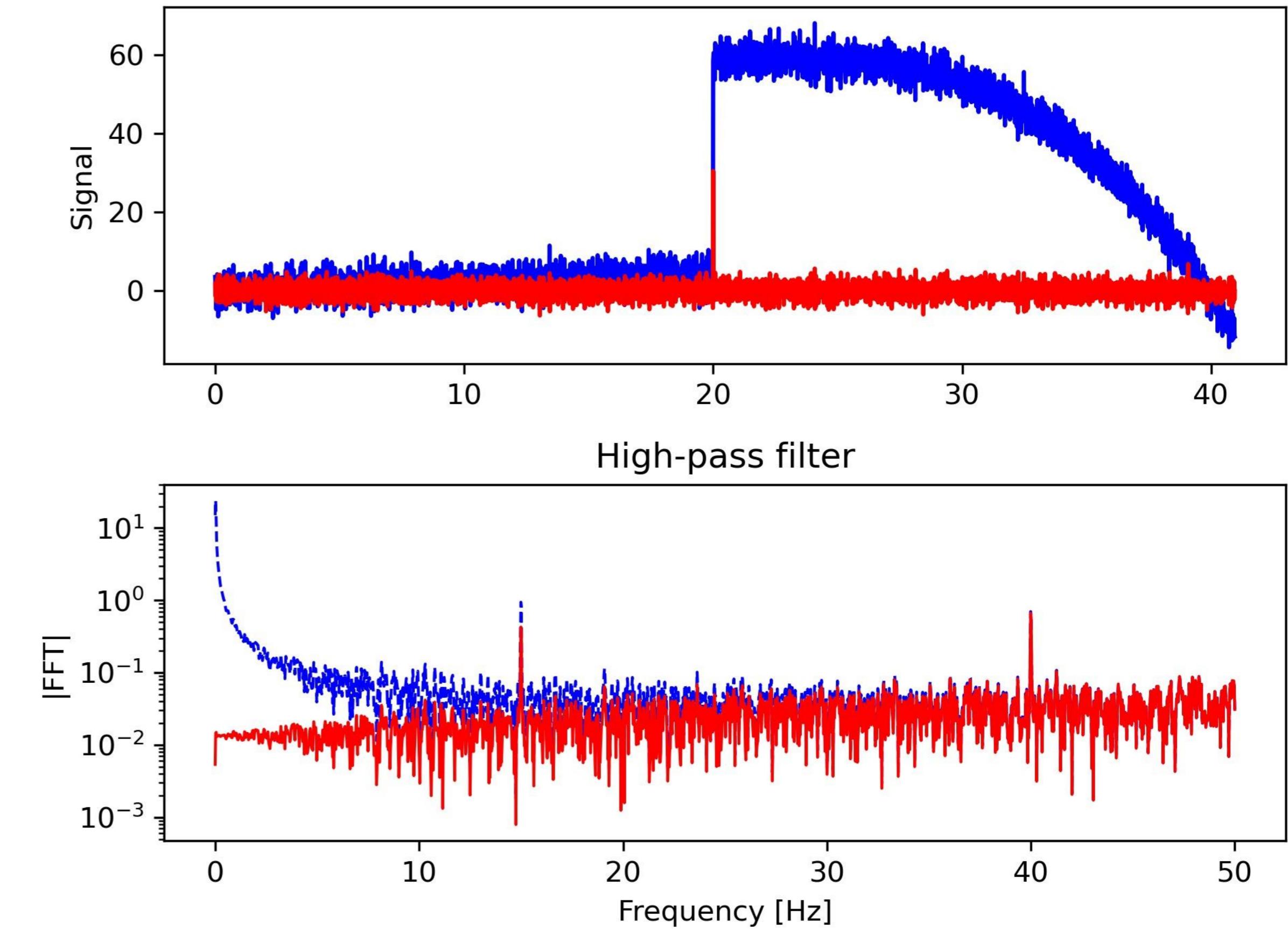
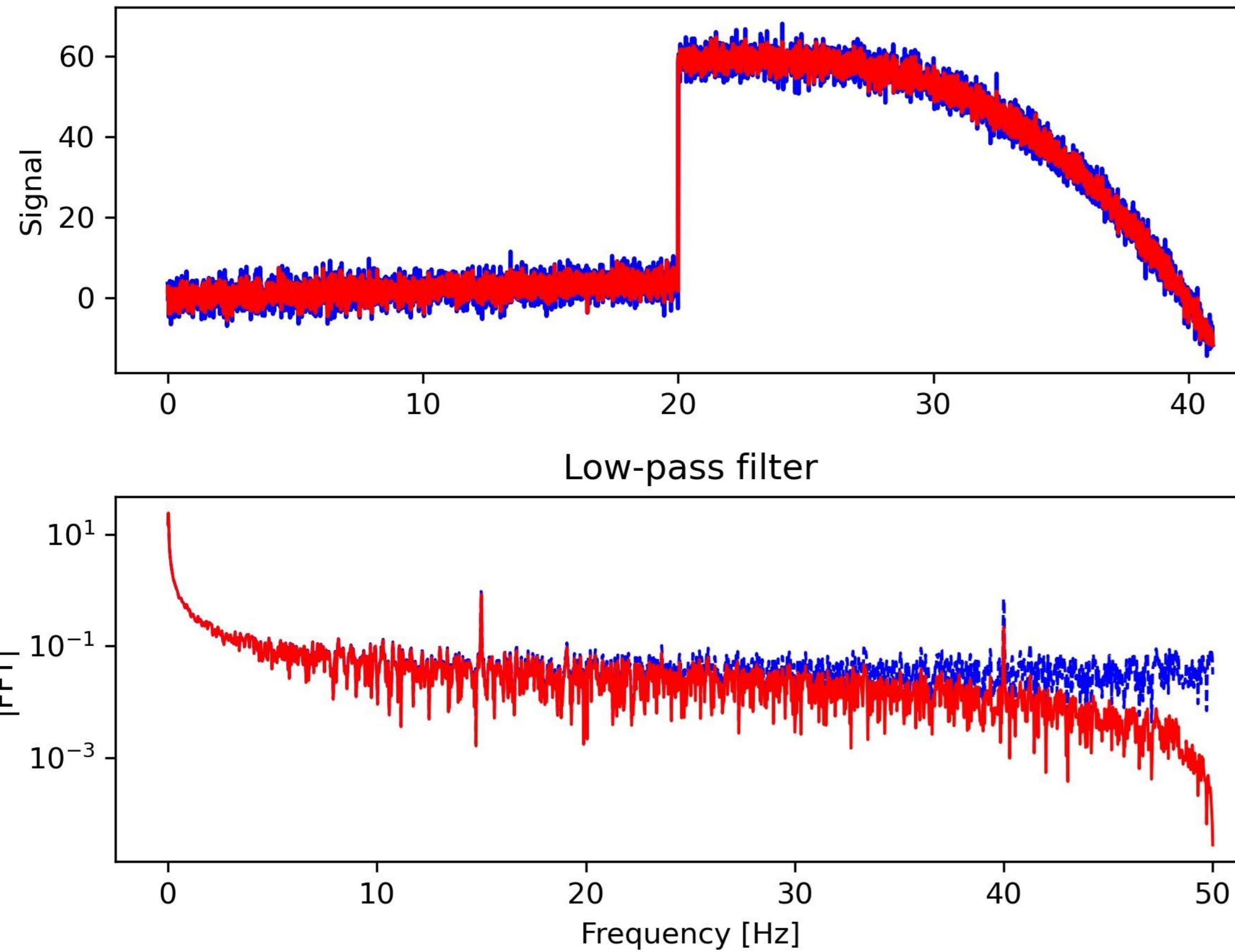
```
[b,a] = signal.butter(ij,fc/(Fs/2),'high');  
w, histbutter = signal.freqz(b,a)
```

```
b,a] = signal.butter(ij,np.array([fc1,fc2])/(Fs/2),'bandpass');  
w, histbutter = signal.freqz(b,a)
```



Filtering

Butterworth

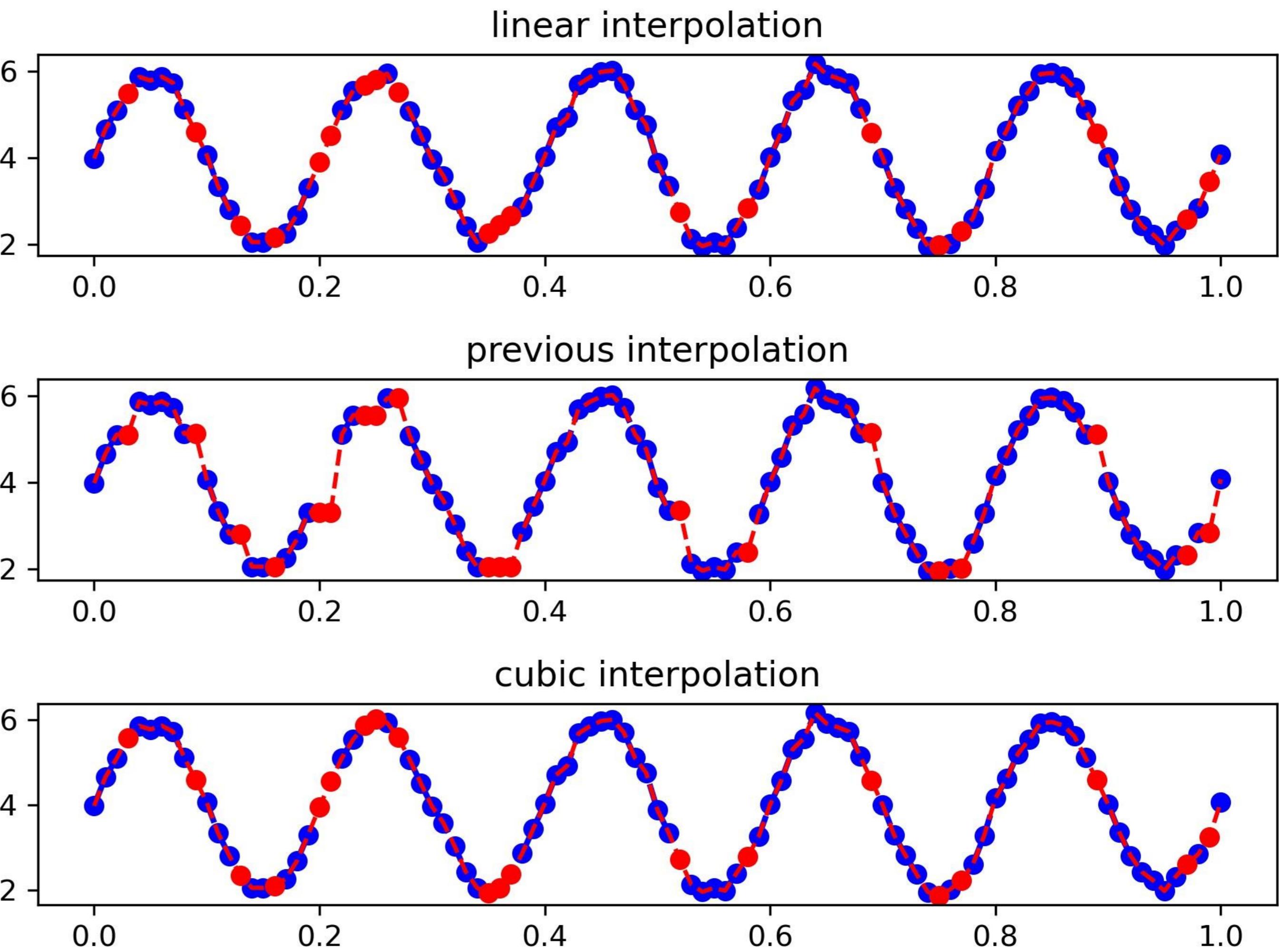
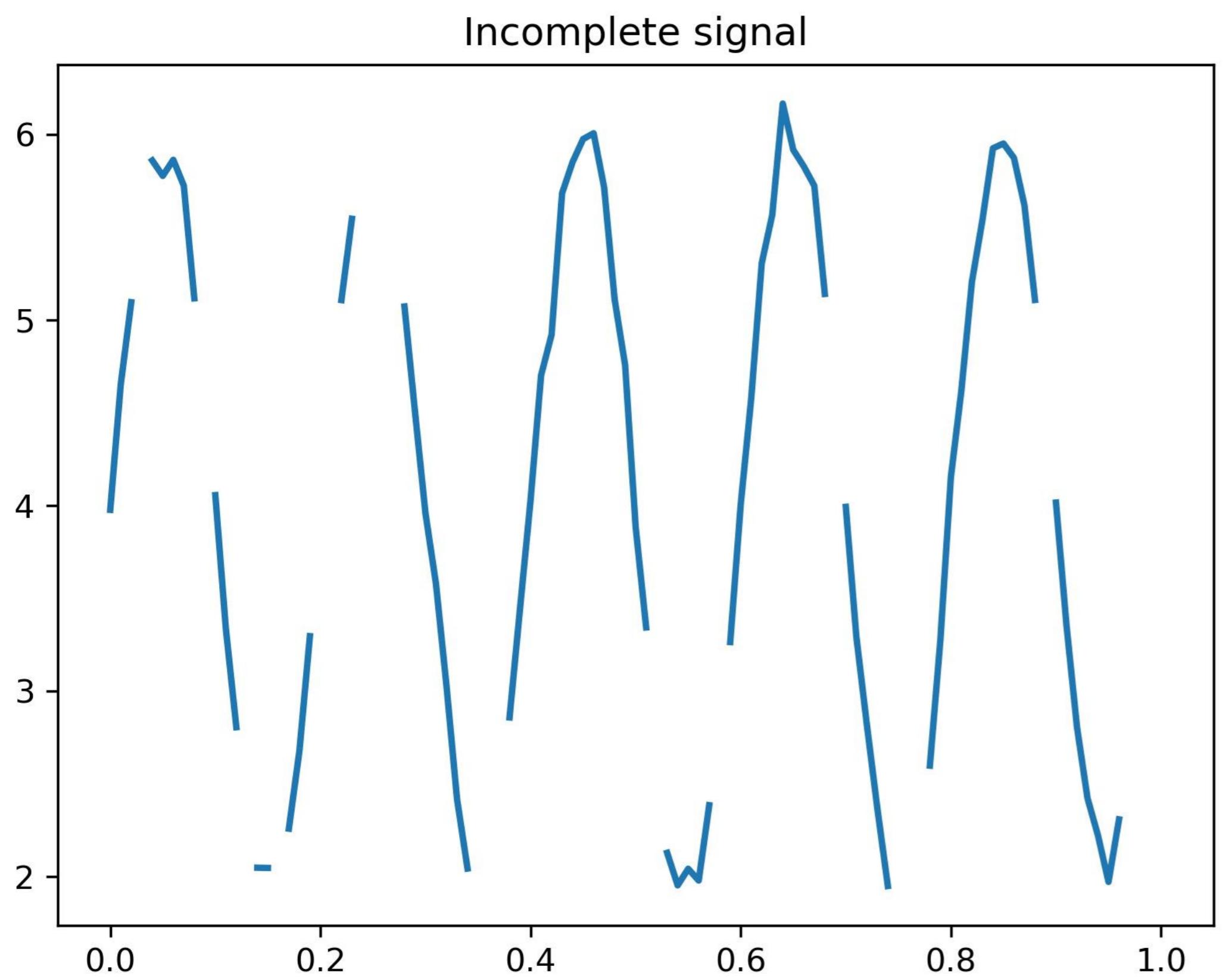




Missing data



Missing data





UNIVERSIDAD
DE GRANADA

Power Spectral Density

Densidad Espectral de Potencia (Power Spectral Density)

Dado: un registro finito de una señal.

Determina: la distribución de la frecuencia de la señal sobre la frecuencia.

Igualdad de Parseval

$$\sum_{t=-\infty}^{\infty} |y(t)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) d\omega$$

Energy Spectral Density

$$S(\omega) = |Y(\omega)|^2$$

Power Spectral Density (PSD)

Primera definición:

$$S(\omega) = \sum_{k=-\infty}^{\infty} r(k) e^{-i\omega k}; \quad r(k) = E\{y(t)y^*(t-k)\}$$

$$r(0) = E\{|y(t)|^2\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) d\omega$$

Potencia de señal infinitesimal
en la banda $\omega + d\omega$



Densidad Espectral de Potencia (Power Spectral Density)

Dado: un registro finito de una señal.

Determina: la distribución de la frecuencia de la señal sobre la frecuencia.

Igualdad de Parseval

$$\sum_{t=-\infty}^{\infty} |y(t)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) d\omega$$

Power Spectral Density (PSD)

Conservación de energía:
La energía en el espacio temporal es igual a la energía en el espacio espectral.

Segunda definición:

$$S(\omega) = \lim_{N \rightarrow \infty} E \left\{ \frac{1}{N} \left| \sum_{t=1}^N y(t) e^{-i\omega t} \right|^2 \right\} = \lim_{N \rightarrow \infty} E \left\{ \frac{1}{N} |Y_N(\omega)|^2 \right\}$$



Power Spectral Density

PROBLEMA: Un registro finito de una señal $y(x)$
Encontrar una estimación de $\hat{S}(\omega)$

Método no-paramétrico: **PERIODOGRAMA (PERIODOGRAM)**

$$S(\omega) = \lim_{N \rightarrow \infty} E \left\{ \frac{1}{N} \left| \sum_{t=1}^N y(t) e^{-i\omega t} \right|^2 \right\} = \lim_{N \rightarrow \infty} E \left\{ \frac{1}{N} |Y_N(\omega)|^2 \right\}$$

$$\hat{S}(\omega) = \frac{1}{N} \left| \sum_{t=1}^N y(t) e^{-i\omega t} \right|^2$$

L_Periodogram.py

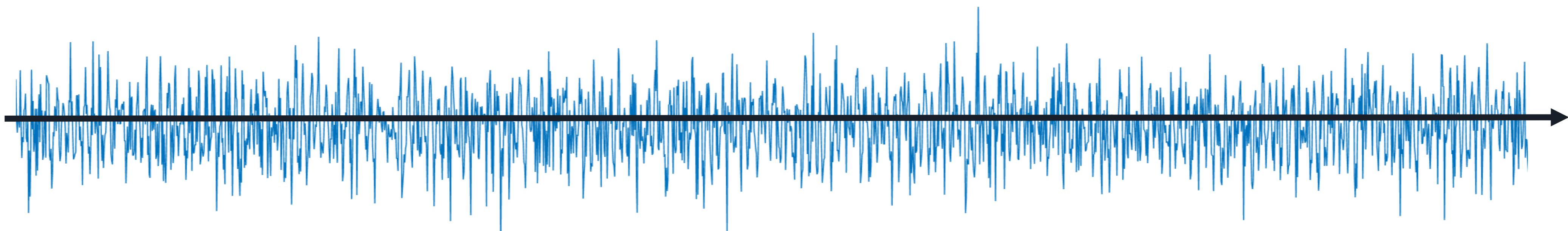
```
from scipy.signal import periodogram
fxx, pxx = periodogram(x,Fs,nfft=np.size(x),window=np.ones(np.size(x)),return_onesided=1,scaling='spectrum', detrend=False)
```



UNIVERSIDAD
DE GRANADA

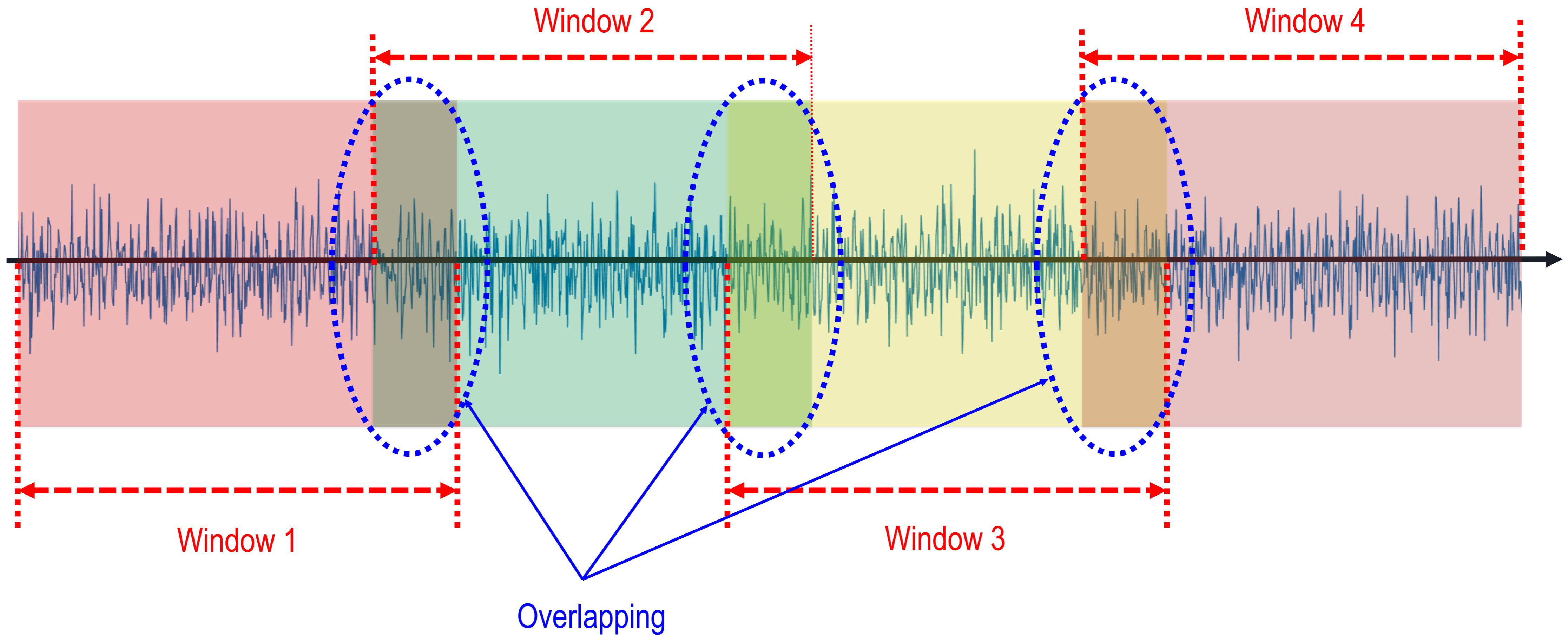
Power Spectral Density

Método paramétrico – **Welch's method**



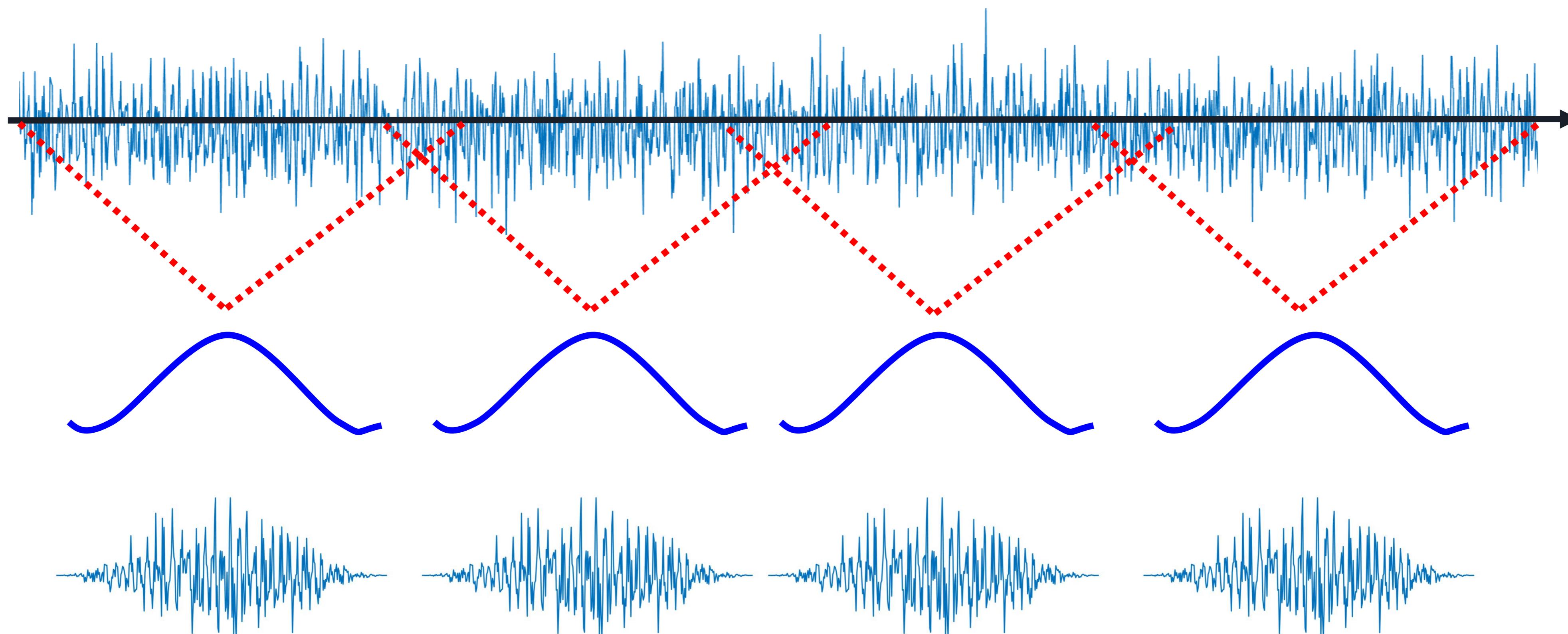
Power Spectral Density

Método paramétrico – Welch's method



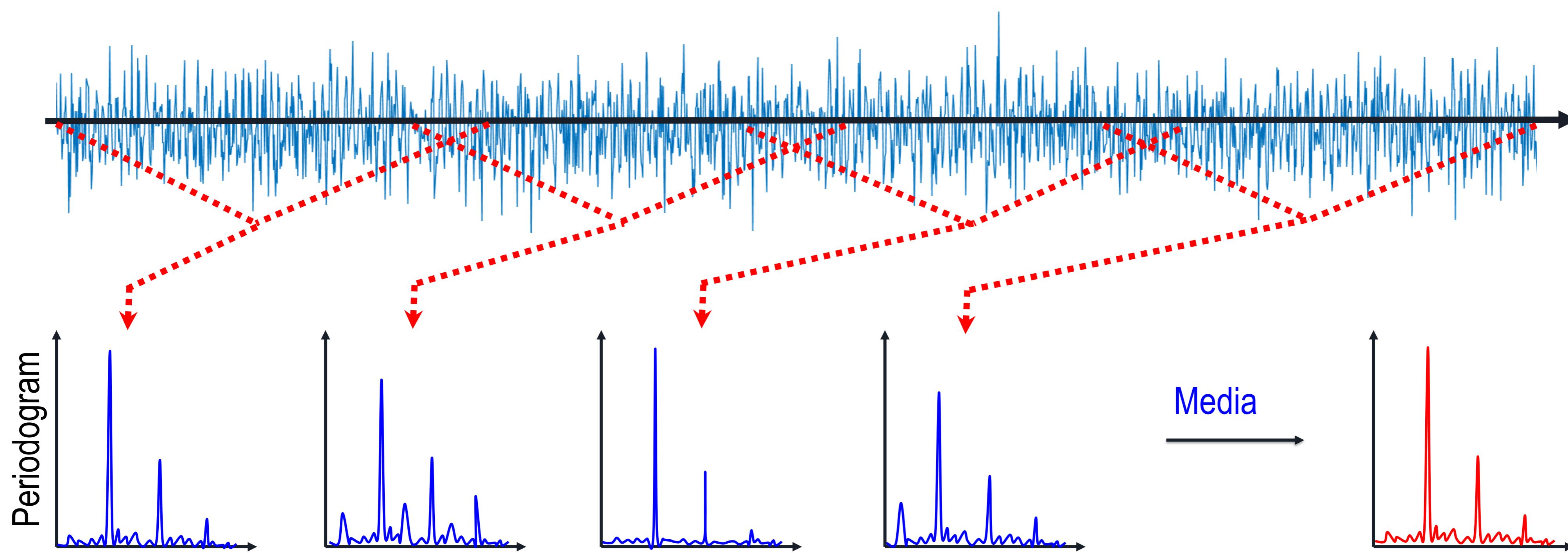
Power Spectral Density

Método paramétrico – **Welch's method**



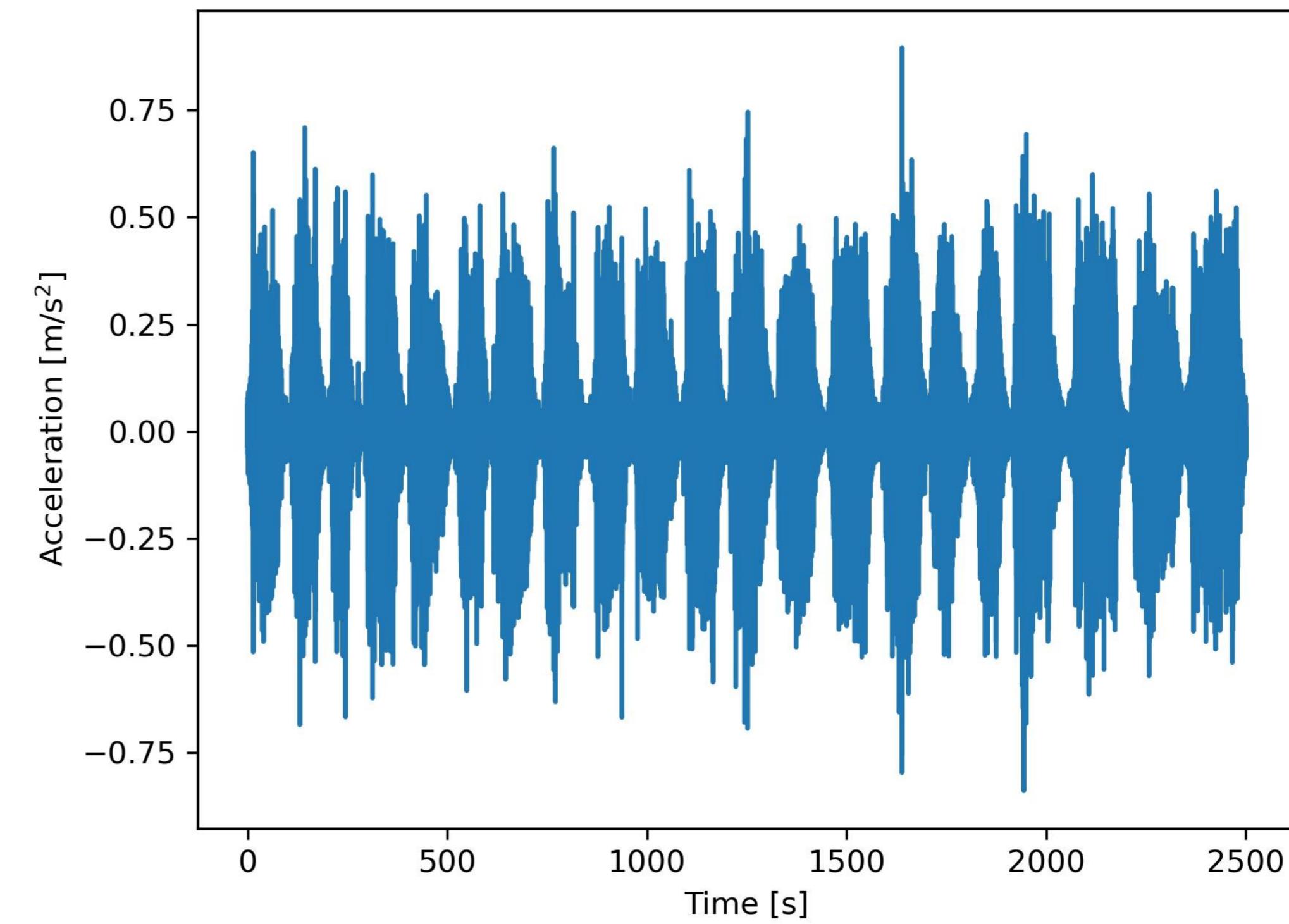
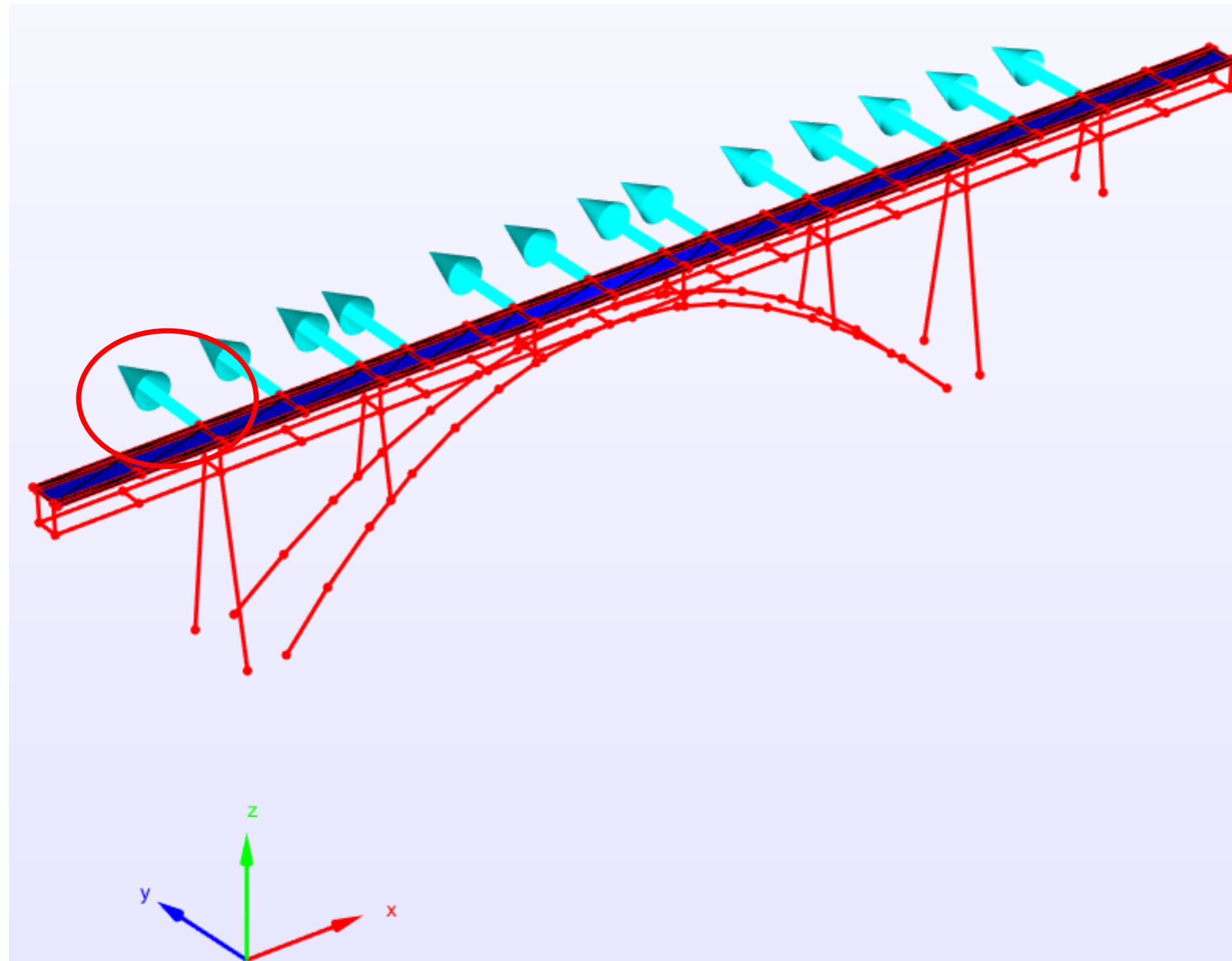
Power Spectral Density

Método paramétrico – Welch's method



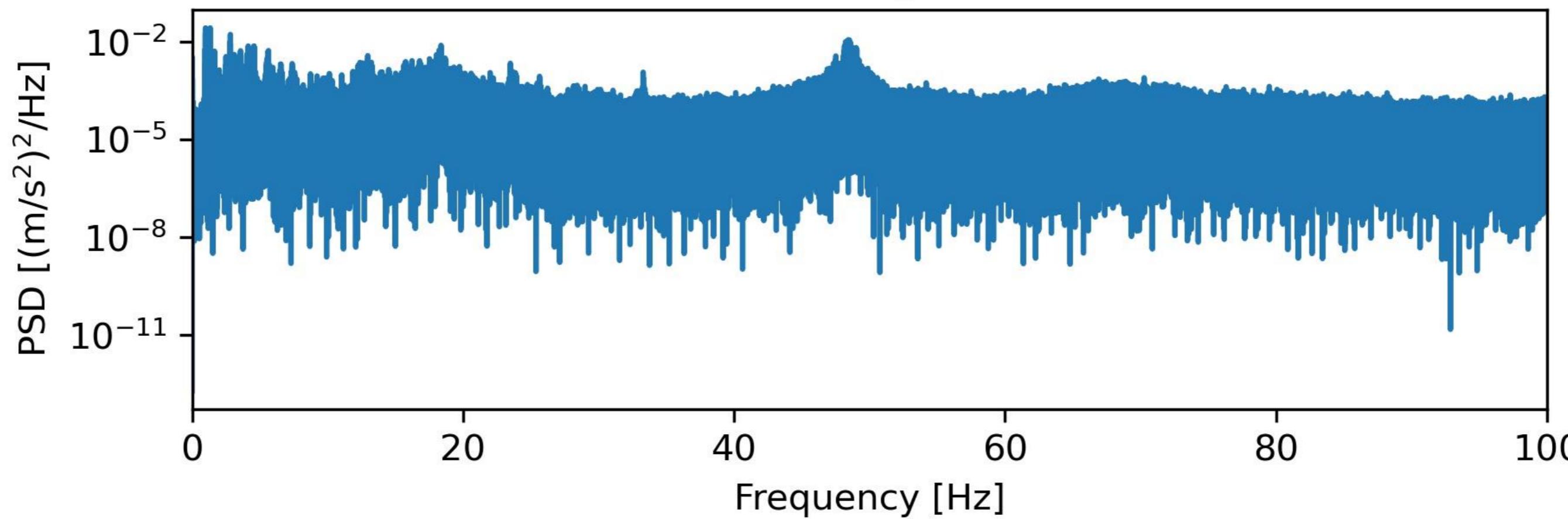
Power Spectral Density

Ejemplo: medición real en Puente ferroviario

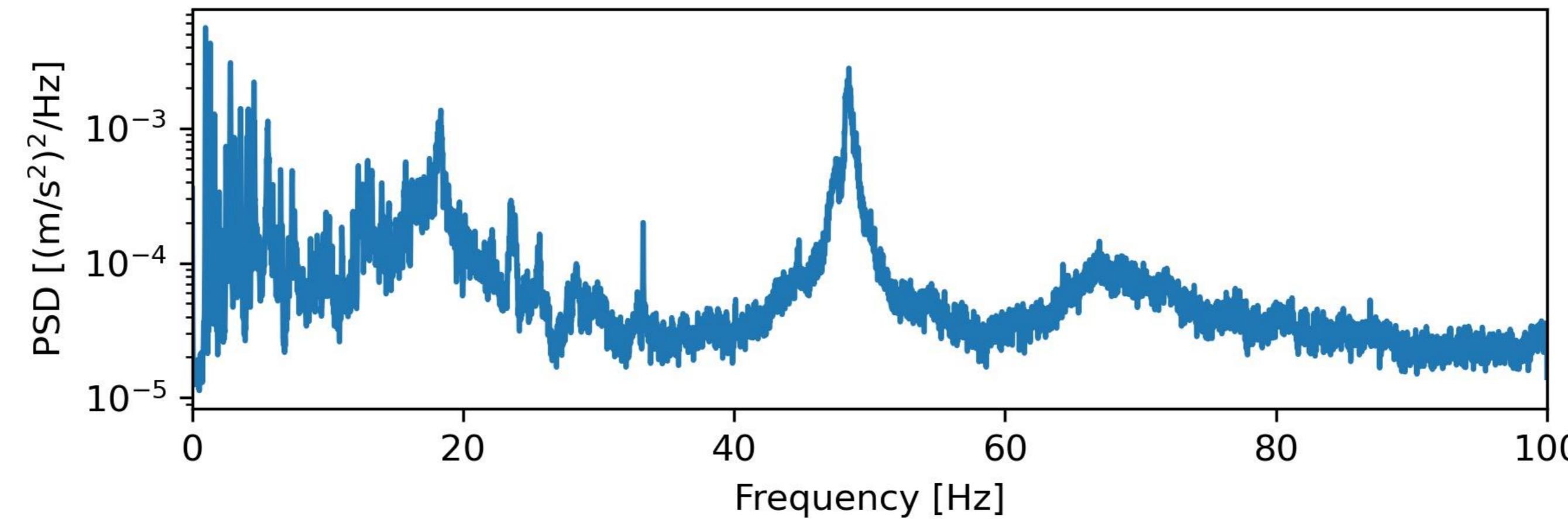


Power Spectral Density

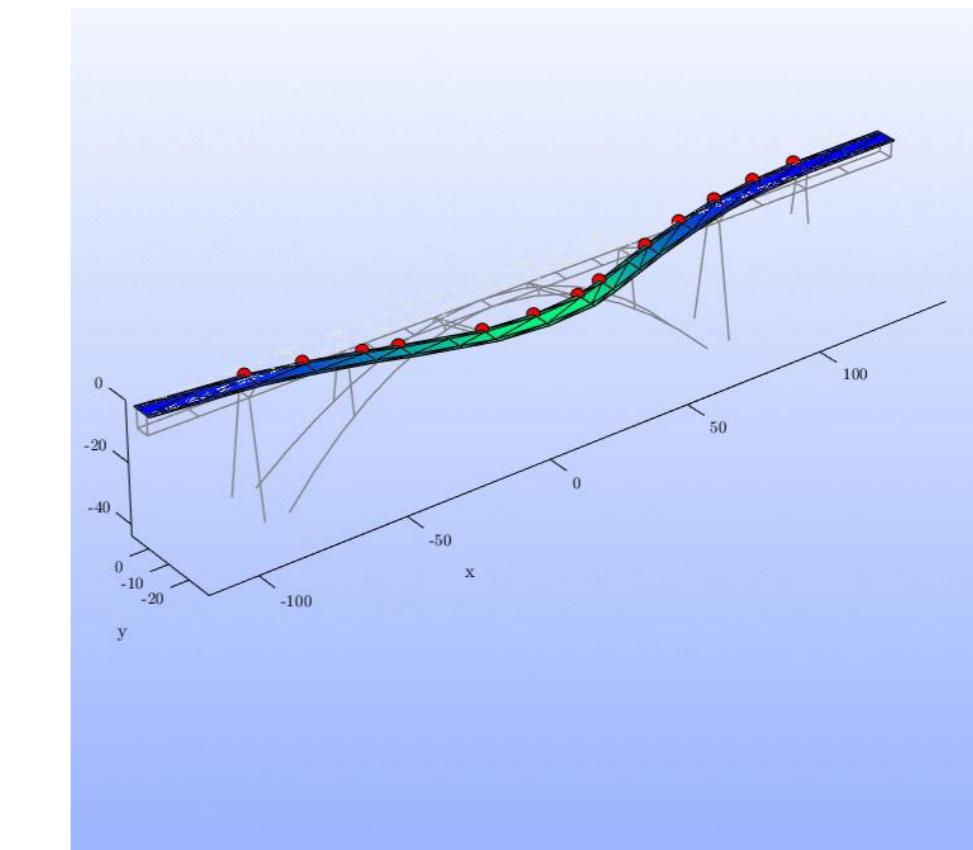
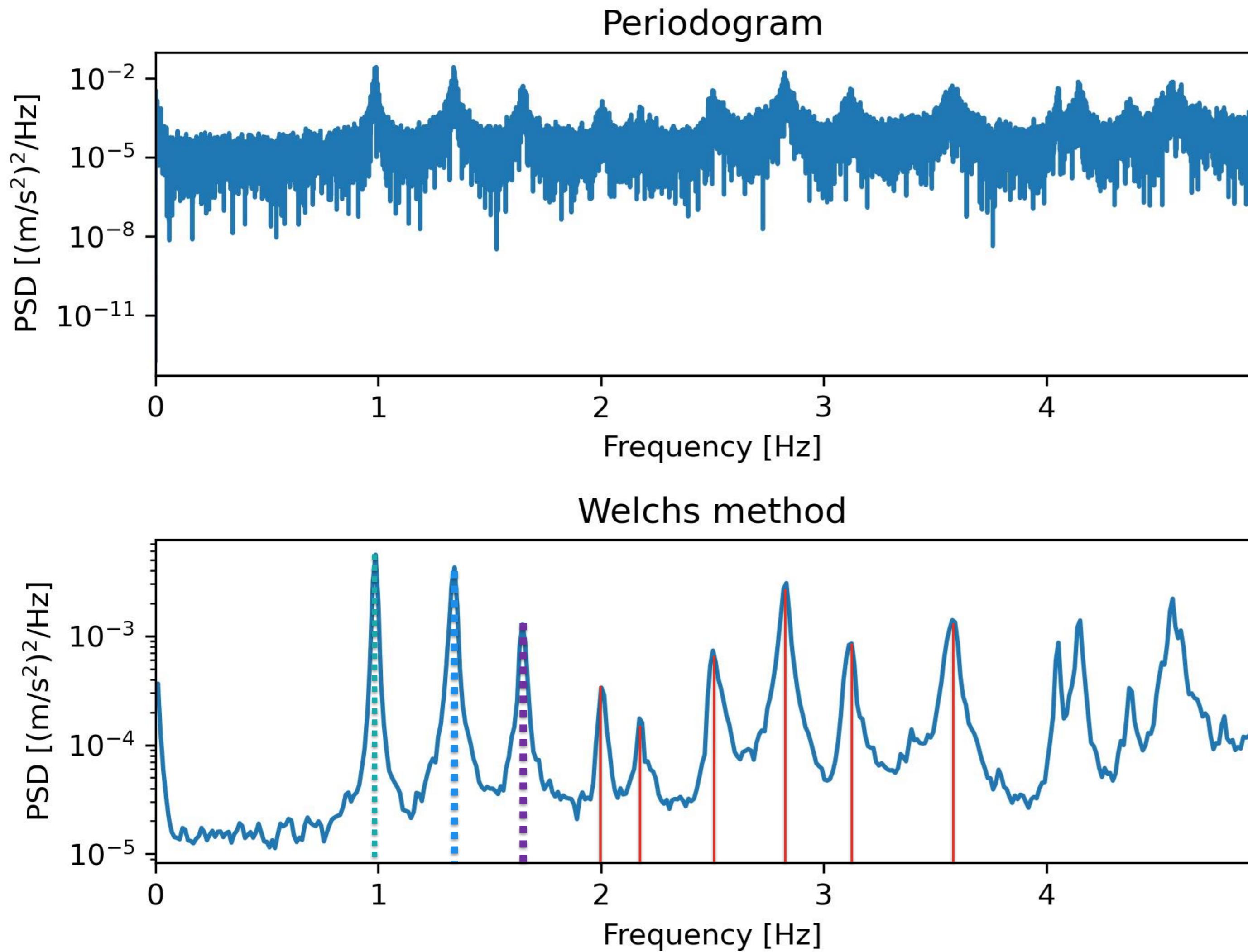
Periodogram



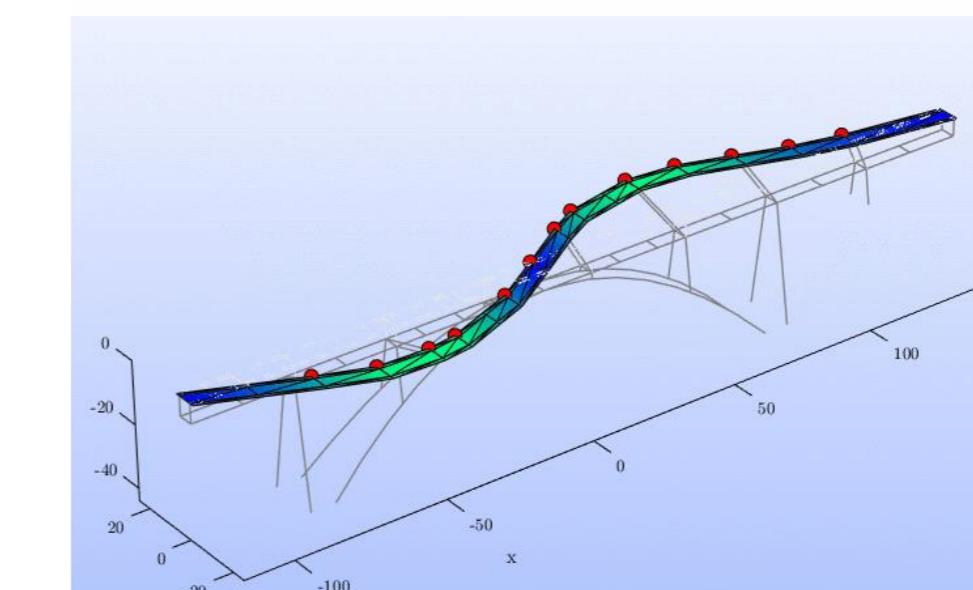
Welchs method



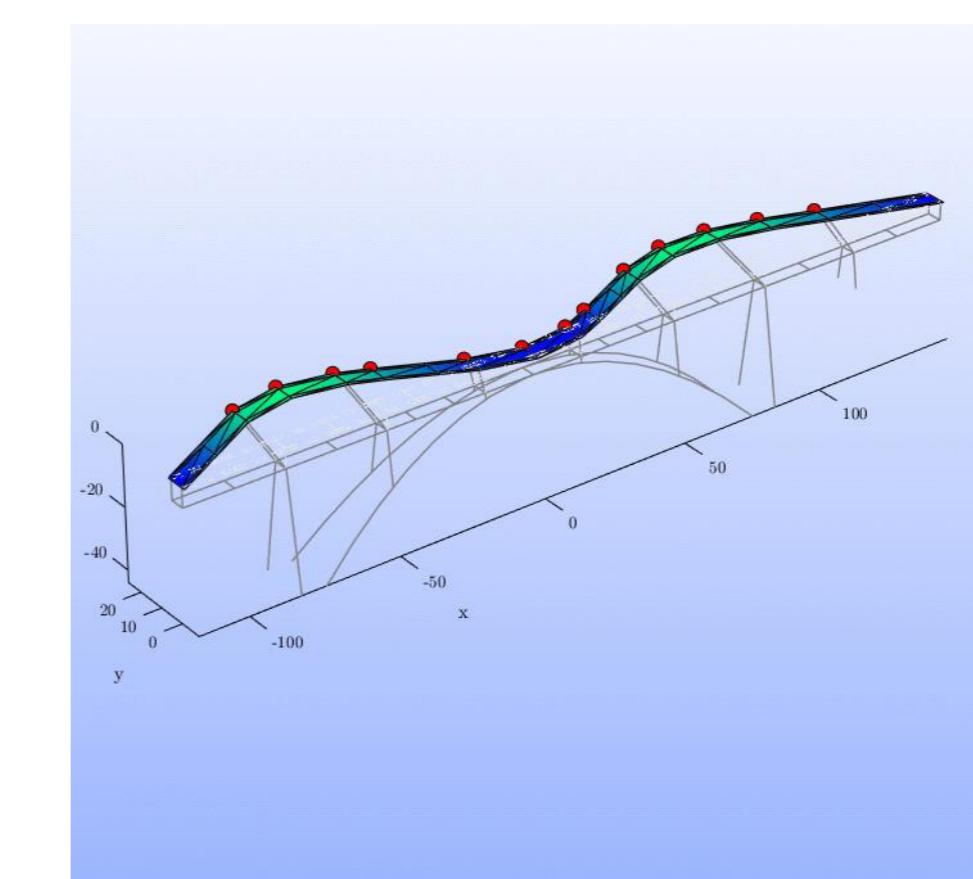
Power Spectral Density



Modo 1



Modo 2



Modo 3

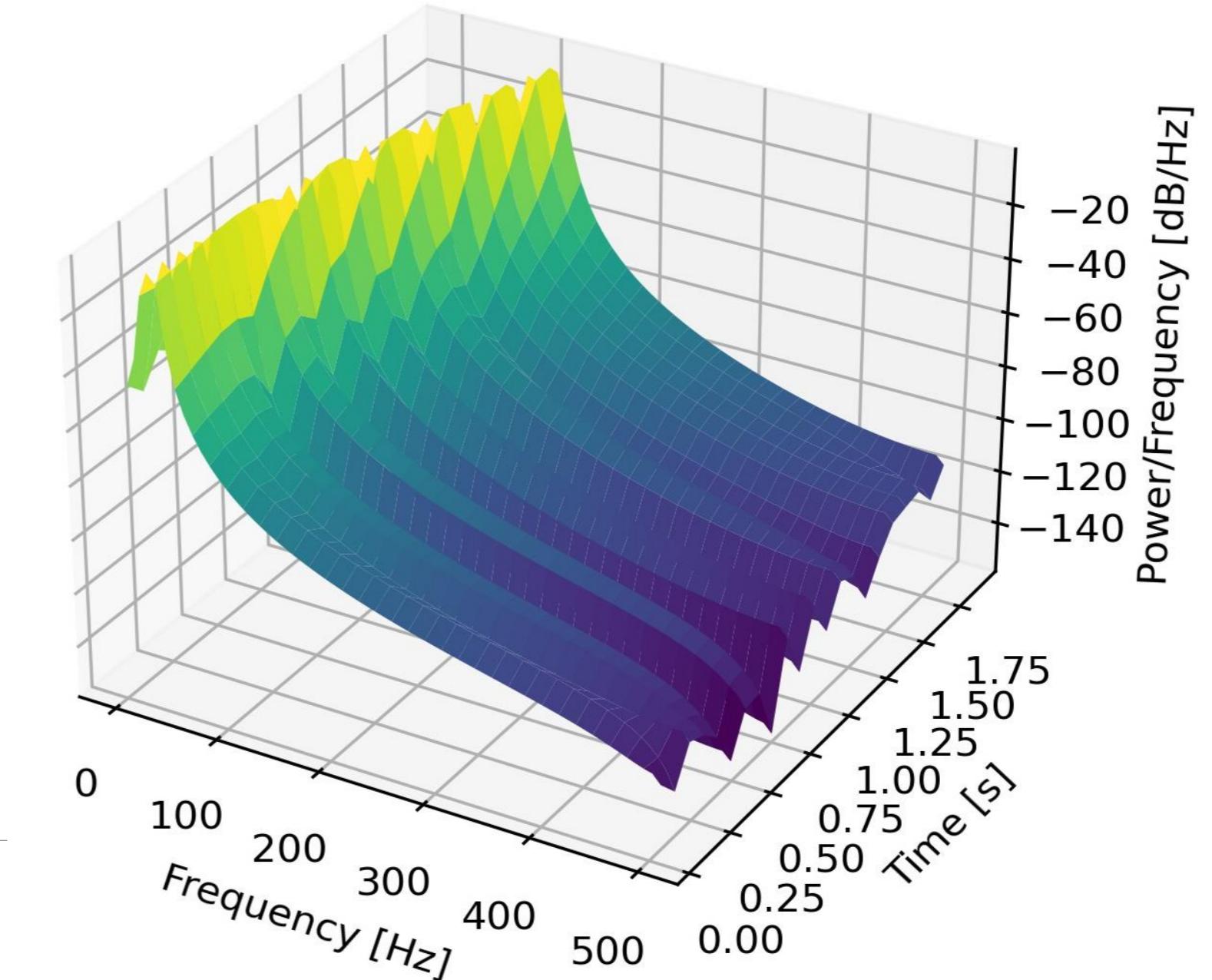
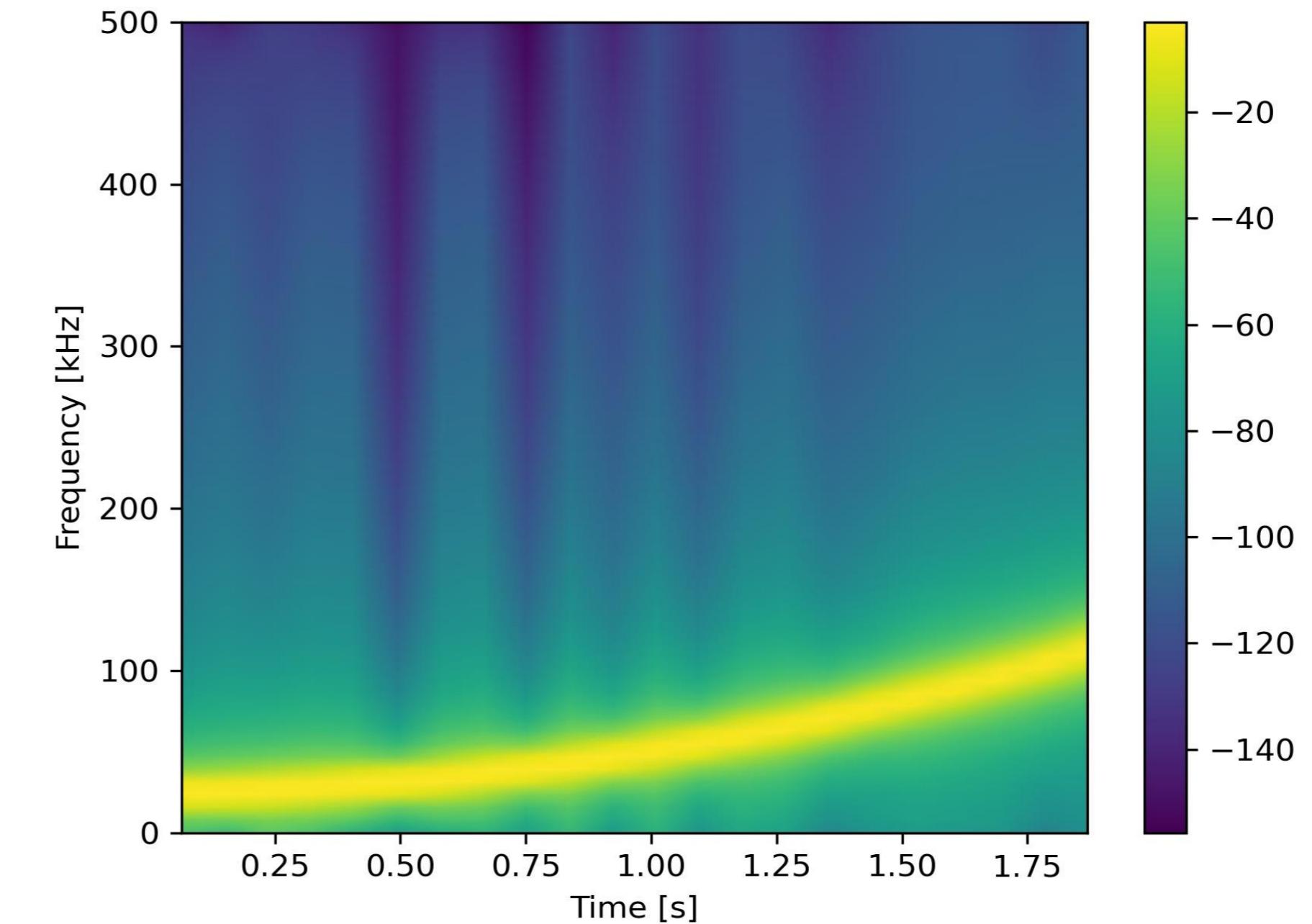
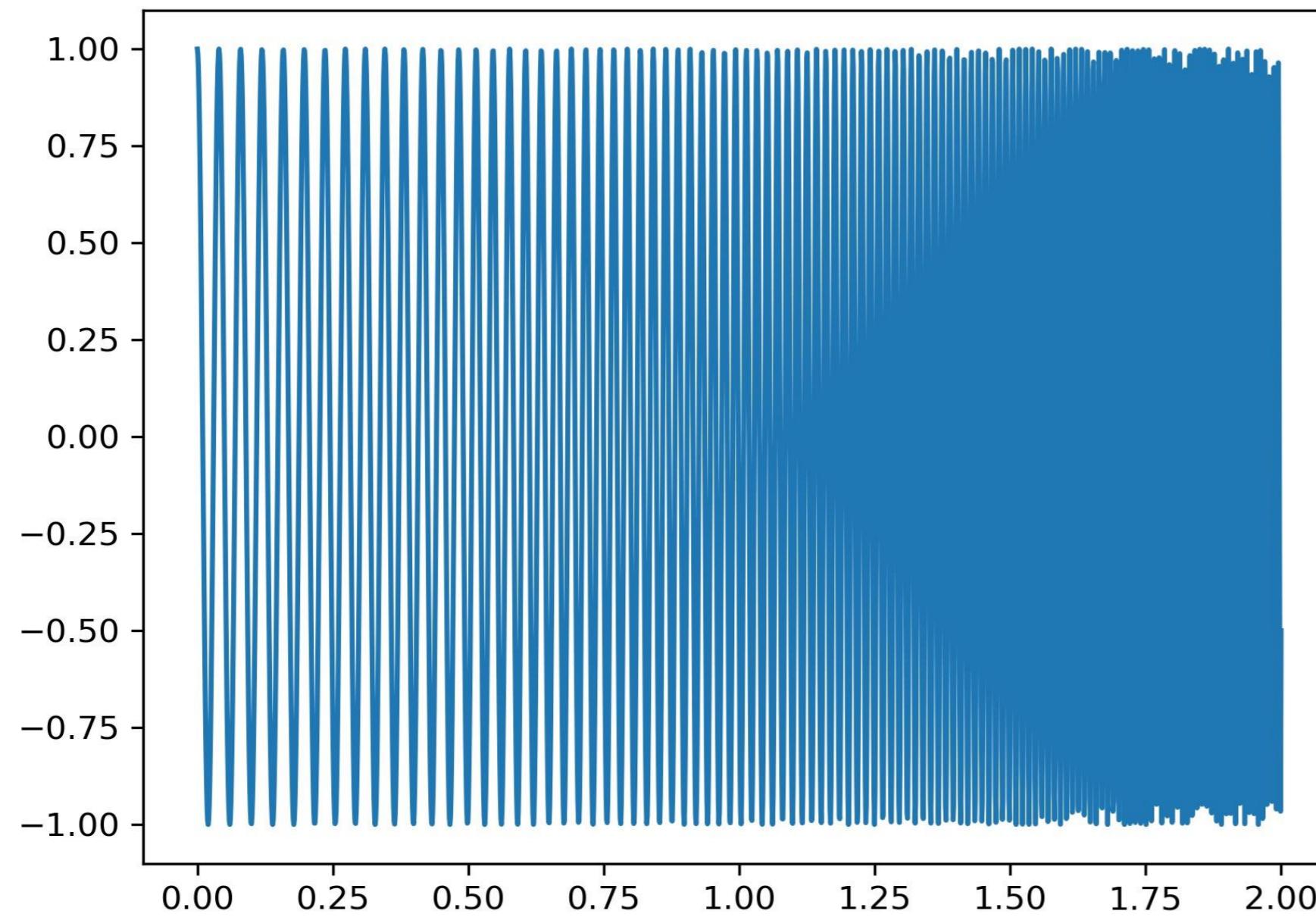




Time-Frequency domain



Time-Frequency domain



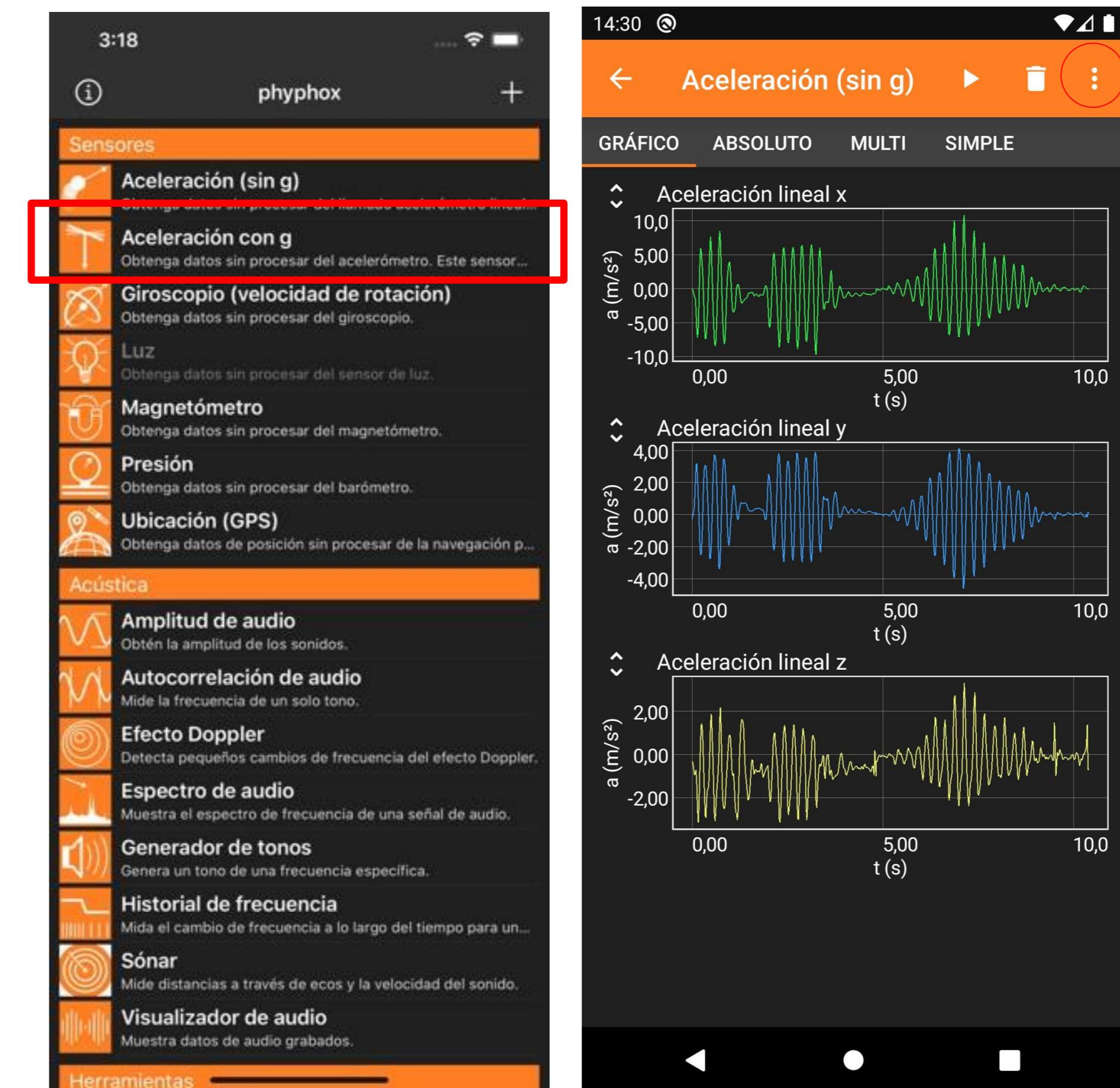


UNIVERSIDAD
DE GRANADA

Práctica MEMS

Práctica MEMS

Phyphox – Aplicación móvil que permite monitorizar todos los sensores de nuestro dispositivo móvil.



Exportar datos

Objetivo

Generar una señal armónica de frecuencia 3 Hz (3 ciclos por segundo)

Práctica MEMS

