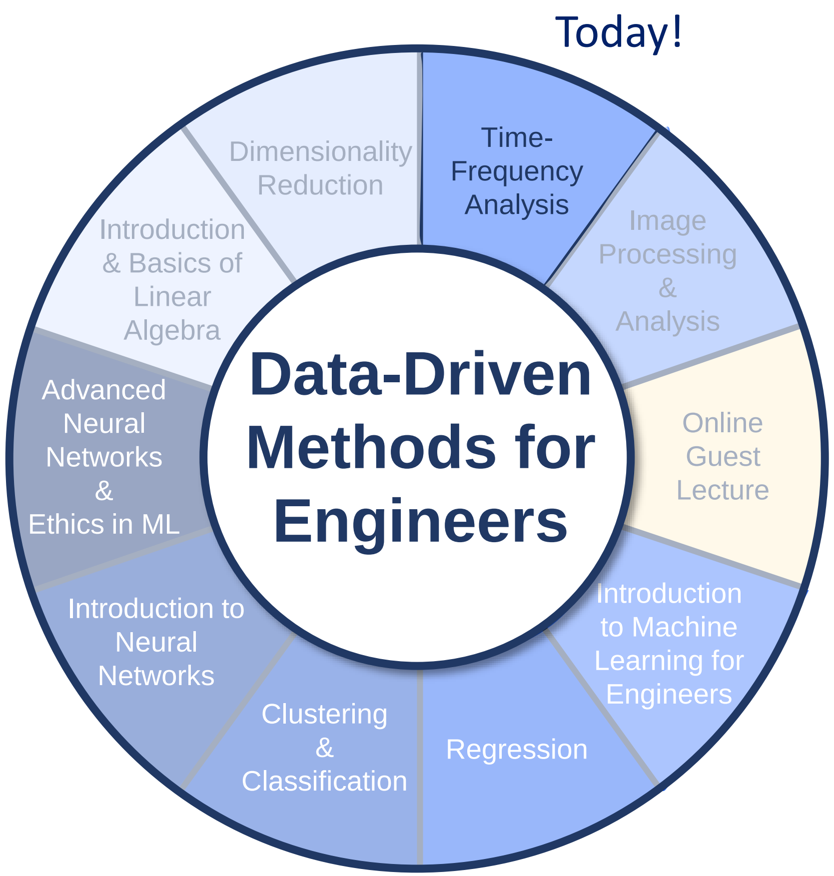
Data-Driven Methods for Engineers (MECH0107) 2025 - 2026

Dr Lama Hamadeh
Module Lead
Office 429 | Roberts Building
Mechanical Engineering Department
l.hamadeh@ucl.ac.uk

Dr Llewellyn Morse
Module Tutor
Office 503D| Roberts Building
Mechanical Engineering Department
l.morse@ucl.ac.uk

Module Lectures



Introduction

- Time-frequency analysis provides a powerful framework for studying non-stationary signals whose spectral characteristics evolve over time, by jointly representing temporal and frequency information.

- By capturing how frequency content changes across time, time-frequency methods enable more accurate monitoring, interpretation, and detection of meaningful patterns and anomalies in such signals.

Electrocardiogram Signals (ECG)

Stock-index profiles

Music and Song Analysis

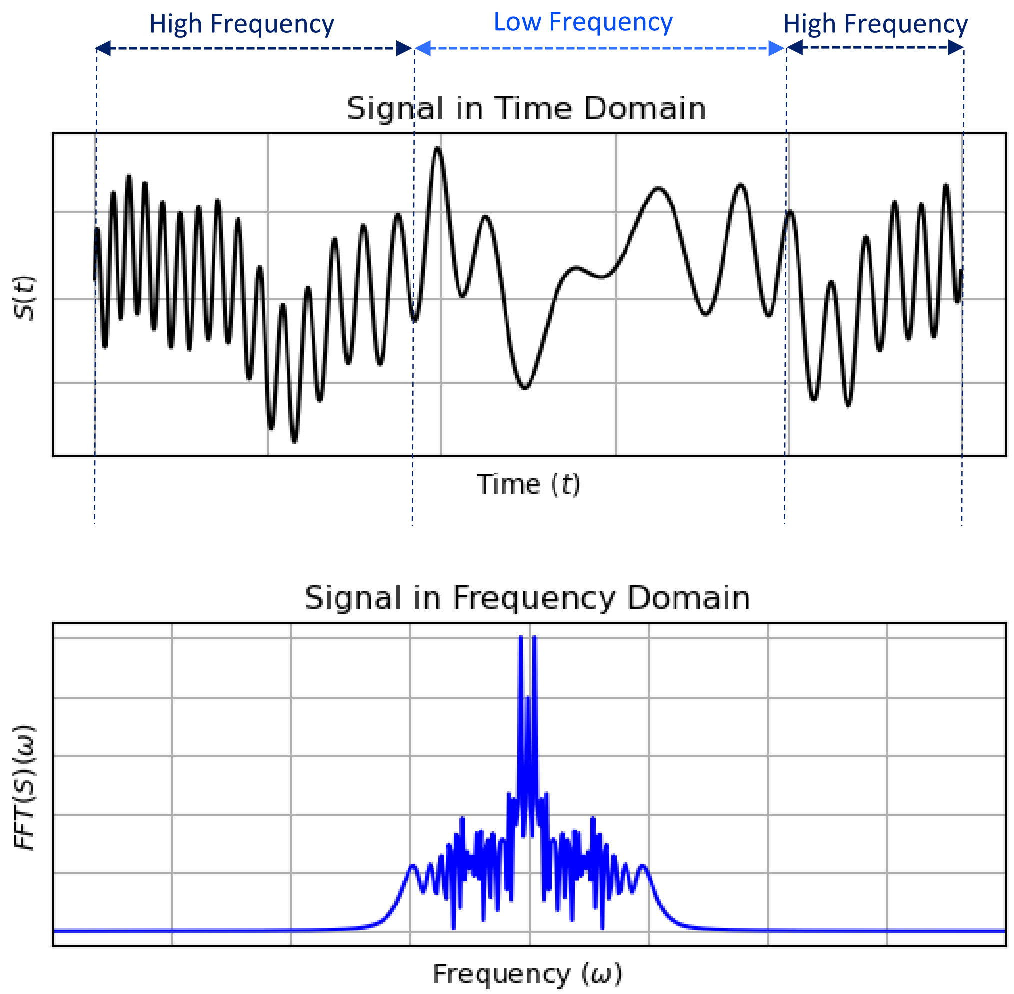
Introduction

- The Fourier transform is one of the most important and foundational methods for the analysis of time signals.

- when transforming a given time signal, it is clear that all the frequency content of the signal can be captured with the transform.

- but there is **<u>no indication</u>** of when the high or low frequencies occur in time!

- **The main limitation** of Fourier Transform is that it fails to capture the moment in time when various frequencies were exhibited.



Fourier Transform (FT)

- Fourier Transform is used to represent a general, non-periodic function by a continuous superposition or integral of complex exponentials.

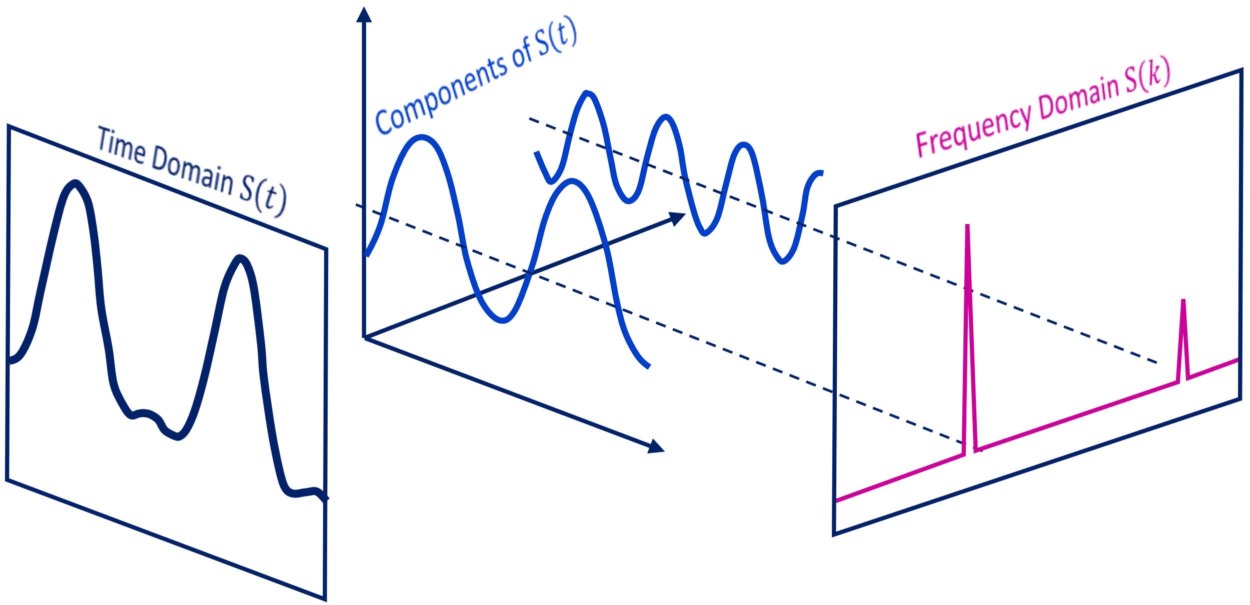
- The Fourier Transform and its inverse are given by:

Frequency

$$F(k) = \int_{-\infty}^{\infty} e^{-ikx} f(x)\, \mathrm{d}x$$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} F(k)\, \mathrm{d}k$$

Fourier Transform pair



What if we need to use this transform on discrete data? How can we implement the infinite integral computationally?

Discrete Fourier Transform (DFT)

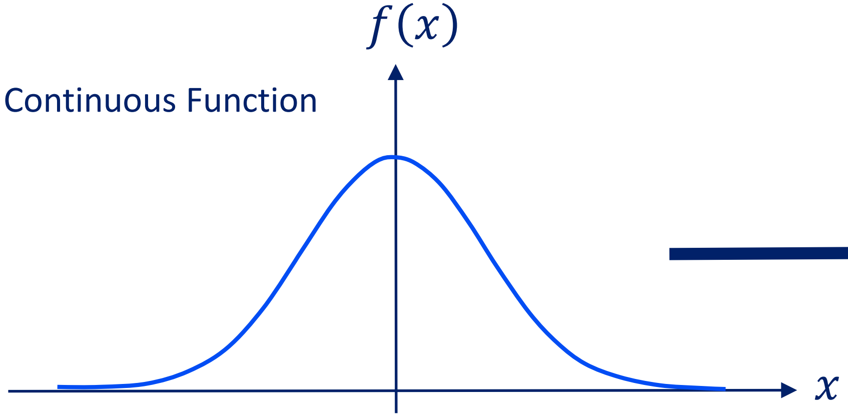
- When working with real data, we don't have analytic function; we have data measurements from experiments or simulations, i.e., **discrete points**. Therefore, it is necessary to approximate the Fourier trans-form on a discrete vector of data.

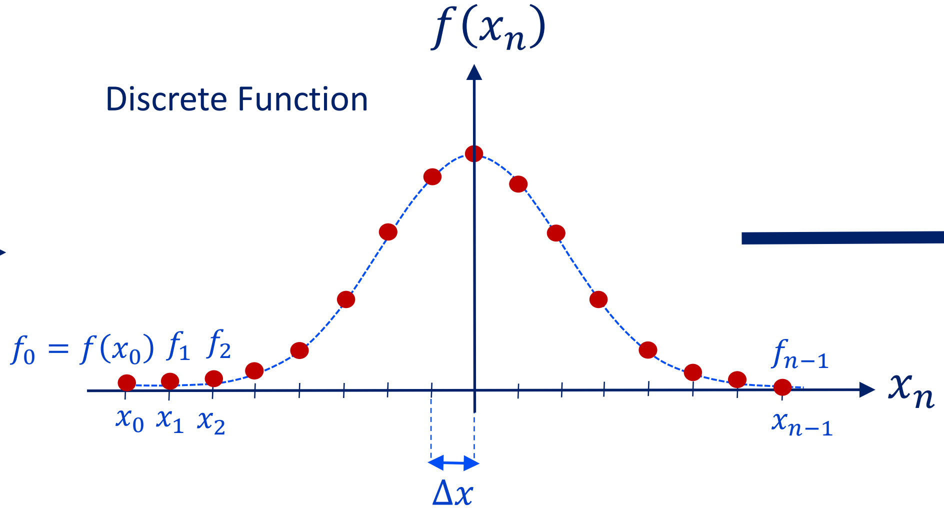
- The resulting truncated form is known as the **Discrete Fourier Transform (DFT)**, which is a mathematical transformation that can be written in terms of a big matrix multiplication.

$$F(k) = \cancel{\int_{-\infty}^{\infty}} e^{-ikx}\, \cancel{f(x)}\, \mathrm{d}x$$

$$\cancel{f(x)} = \frac{1}{2\pi} \cancel{\int_{-\infty}^{\infty}} e^{ikx} F(k)\, \mathrm{d}k$$

Discrete Fourier Transform (DFT)





Vector of all functions

$$\boldsymbol{f} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{n-1} \end{pmatrix}$$

DFT

$$\widehat{f_k} = \sum_{j=0}^{n-1} f_j \, e^{-\frac{i2\pi}{n} jk}$$

$k = 0,1,2, \dots, n-1$

Vector of all DFTs

$$\hat{\boldsymbol{f}} = \begin{pmatrix} \widehat{f_0} \\ \widehat{f_1} \\ \widehat{f_2} \\ \vdots \\ \widehat{f_{n-1}} \end{pmatrix}$$

Discrete Fourier Transform (DFT)

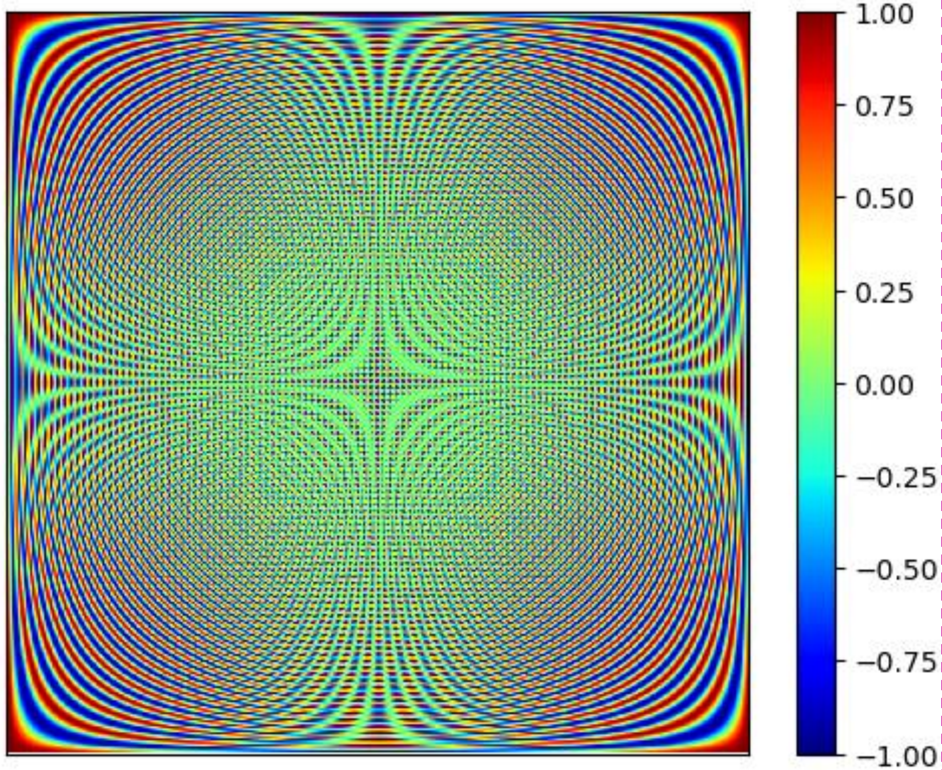
DFT: $$\hat{f}_k = \sum_{j=0}^{n-1} f_j \, e^{-\frac{i2\pi}{n}jk} \qquad \omega_n^{jk}$$

$$\omega_n = e^{-\frac{i2\pi}{n}}$$

Inverse DFT: $$f_j = \frac{1}{n}\sum_{k=0}^{n-1} \hat{f}_k \, e^{\frac{i2\pi}{n}jk}$$

$$\begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_{n-1} \end{bmatrix} = \begin{array}{c|c} & 0 \xrightarrow{\;j\;} \quad n-1 \\ \hline \begin{array}{c} 0 \\ \downarrow k \\ \\ \\ n-1 \end{array} & \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)^2} \end{bmatrix} \end{array} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{n-1} \end{bmatrix}$$

Real Part



+

cos

−

Discrete Fourier Transform (DFT)

$$\hat{f}_k = \sum_{j=0}^{n-1} e^{-\frac{i2\pi}{n}jk} f_j$$

Notice that many terms in the DFT repeat (mainly $e^{-\frac{i2\pi}{n}jk}$).

Can we reuse the results that have already been computed so we can reduce the number of total operations?

For each output, you need n multiplications and additions

$$\begin{array}{c} \\ n \text{ outputs} \downarrow \end{array} \begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)^2} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{n-1} \end{bmatrix}$$

The total operations are

$n \times n = n^2$

Or

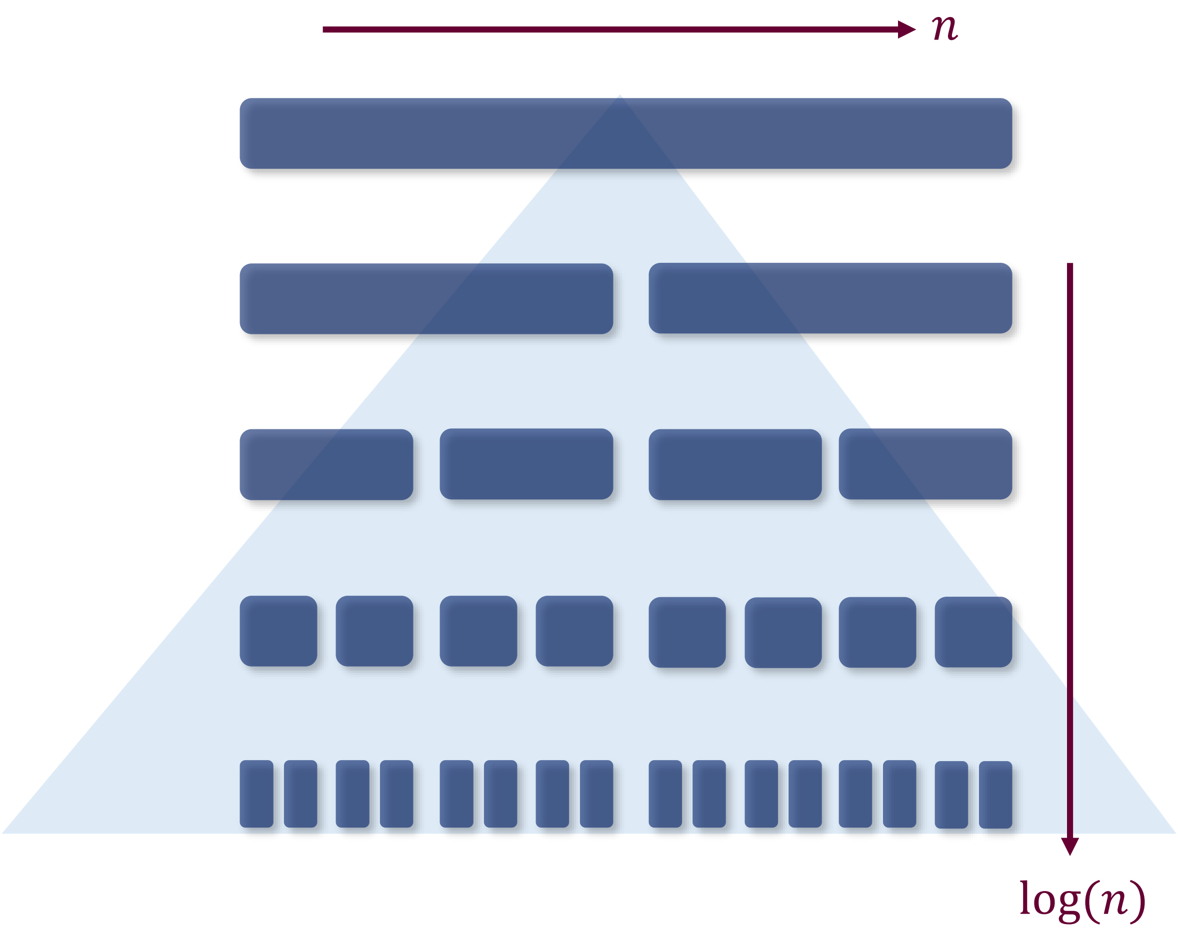
$O(n^2)$

Fast Fourier Transform (FFT)

- In the mid-1960s, Cooley and Tukey developed what is now commonly known as the fast Fourier transform, the FFT algorithm.

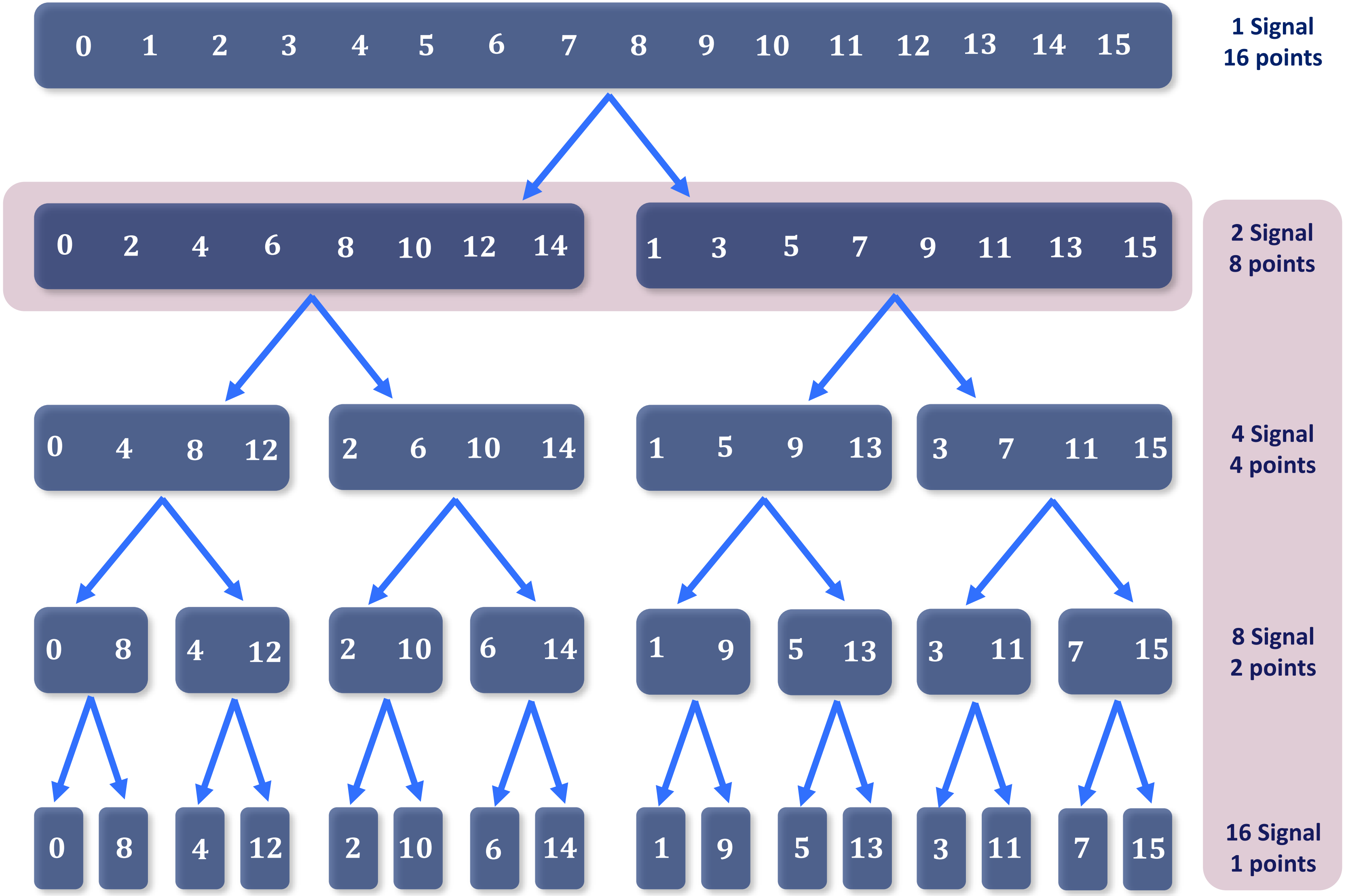
- The FFT is a clever way to compute the Discrete Fourier Transform (DFT) much faster by reusing computations instead of repeating them.

- The FFT breaks the problem into smaller pieces using the so-called: butterfly structure.



Fast Fourier Transform (FFT)

Number of operations is n in every stage



The total operations are $n\log_2(n)$

$\log_2(n)$ stages required in this decomposition, i.e., a 16 point signal (2^4) requires 4 stages

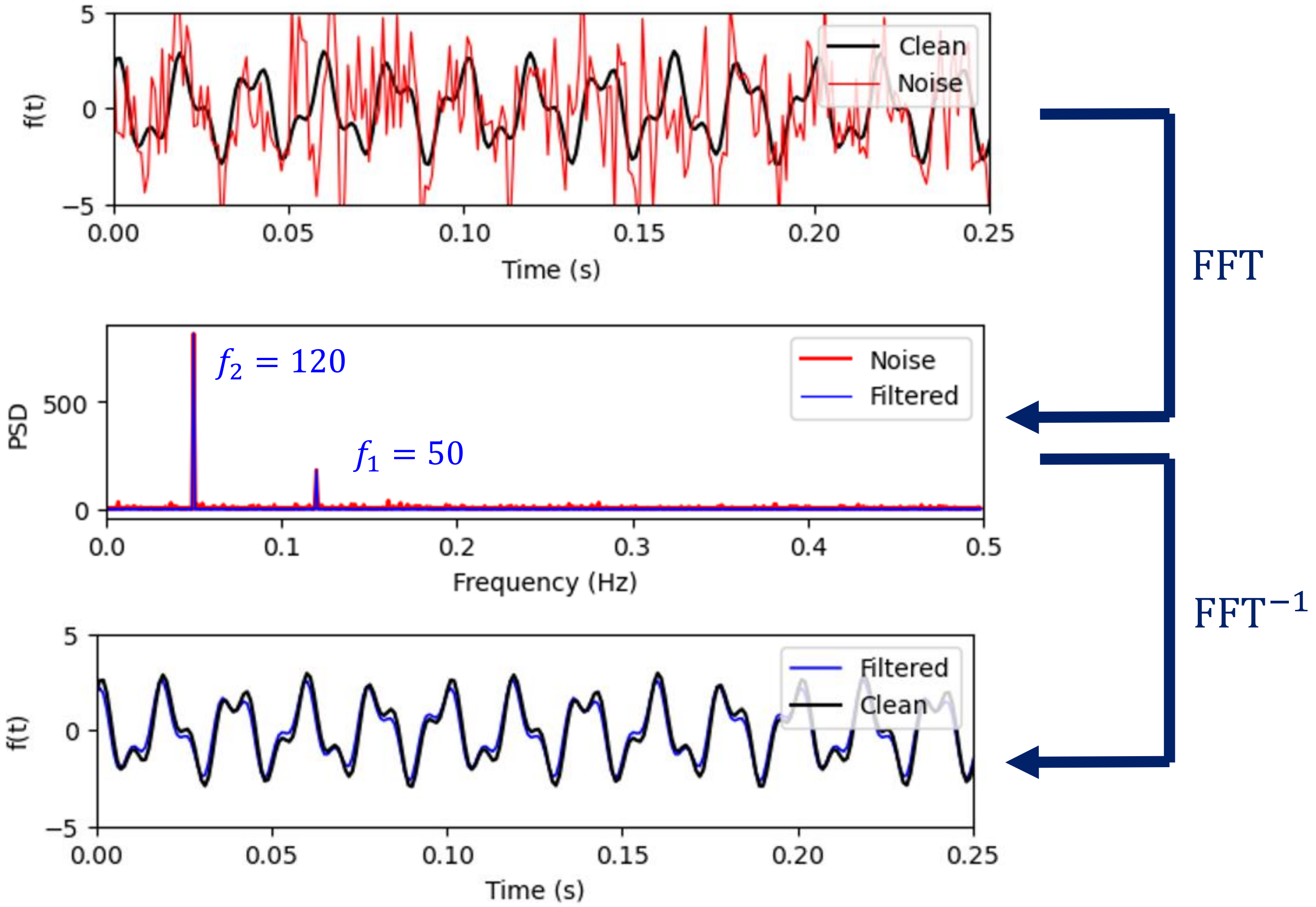
Fast Fourier Transform (FFT)

$$f(t) = 2\cos(2\pi f_1 t) + \sin(2\pi f_2 t)$$

Does FFT provide any information of "***when***" these two frequencies happen?

NO!

Can the FFT be modified so the time information is obtained as well?



Windowed Fourier Transform

- The Hungarian physicist, mathematician, and electrical engineer Gabor Denes (Nobel Prize for Physics in 1971 for the discovery of holography in 1947) was the first to propose a formal method for localizing both time and frequency.

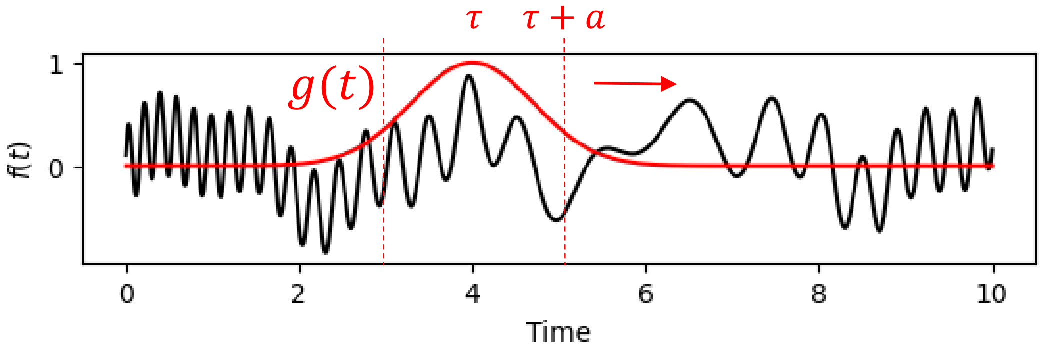
- The Gabor transform, also known as the short-time Fourier transform (STFT), and is defined as:

$$\mathcal{G}[f](t,\omega) = \hat{f}_g(t,\omega) = \int_{-\infty}^{\infty} f(\tau)\underbrace{\bar{g}(t-\tau)e^{-i\omega\tau}}_{\text{Gabor Kernel}}\mathrm{d}\tau$$

$$\boxed{g(t) = e^{-(t-\tau)^2/a^2}}$$

center of the moving window (τ)

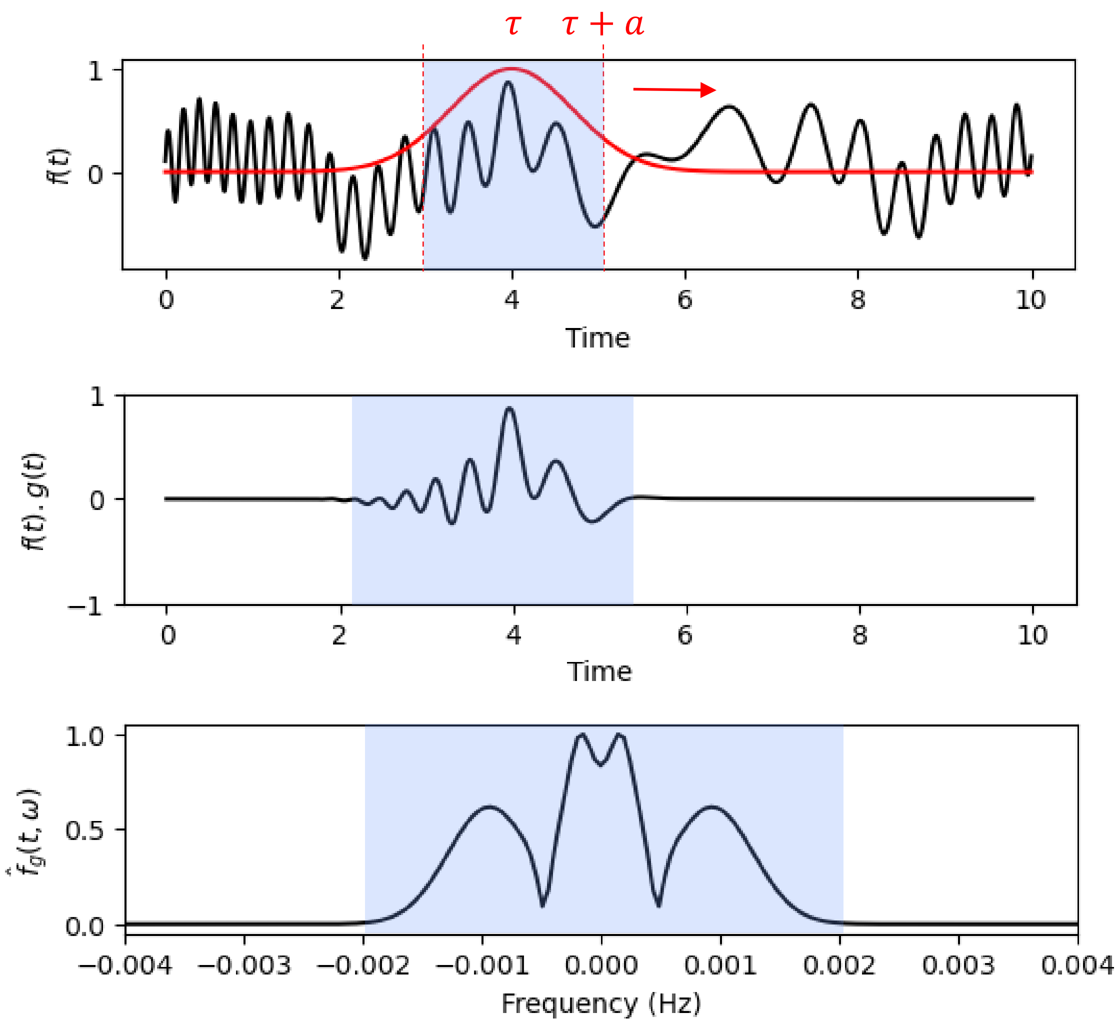
spread (or the width) of the short-time window (a)



Windowed Fourier Transform

- The **broader the window**, the more accurate the frequency content is captured at the sacrifice of less accurate localization of where the signal is in time.

- The **narrower the window**, the more accurate the information on the time domain and the less information on the frequency resolution.

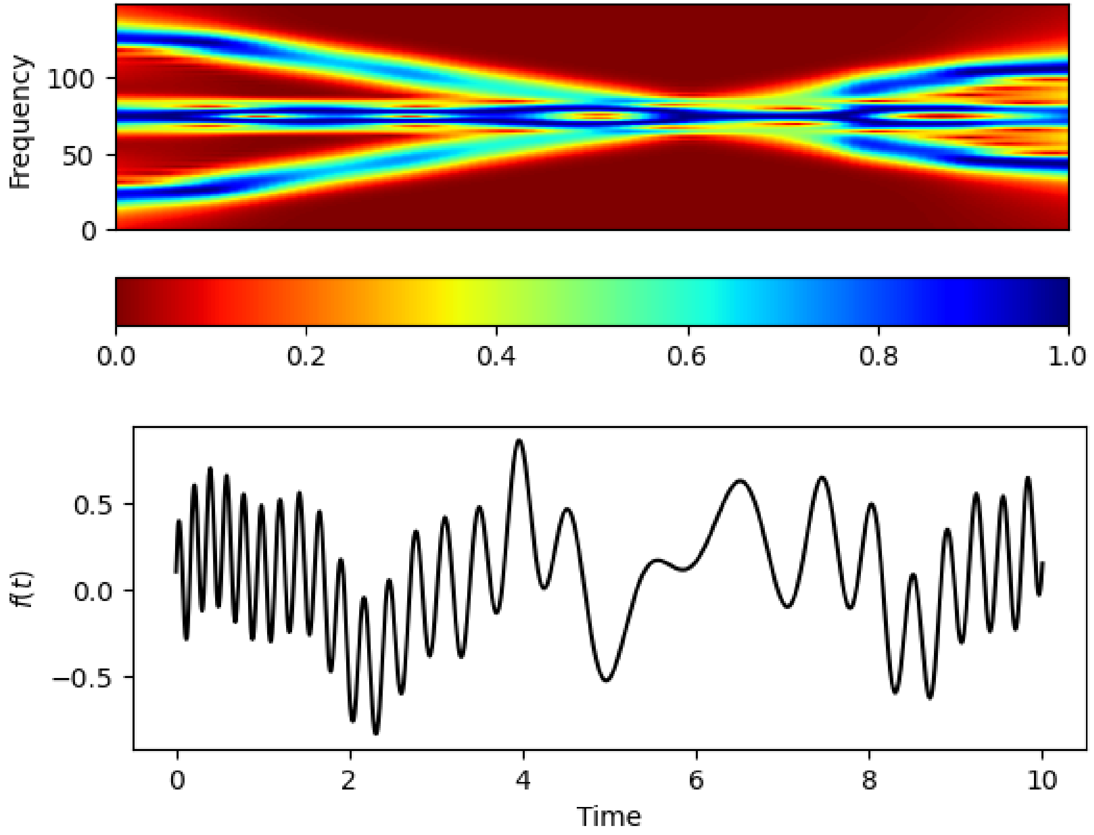


Windowed Fourier Transform

- A spectrogram is a visual representation of the spectrum of frequencies of a signal and their amplitudes as they vary with time, all on one graph.

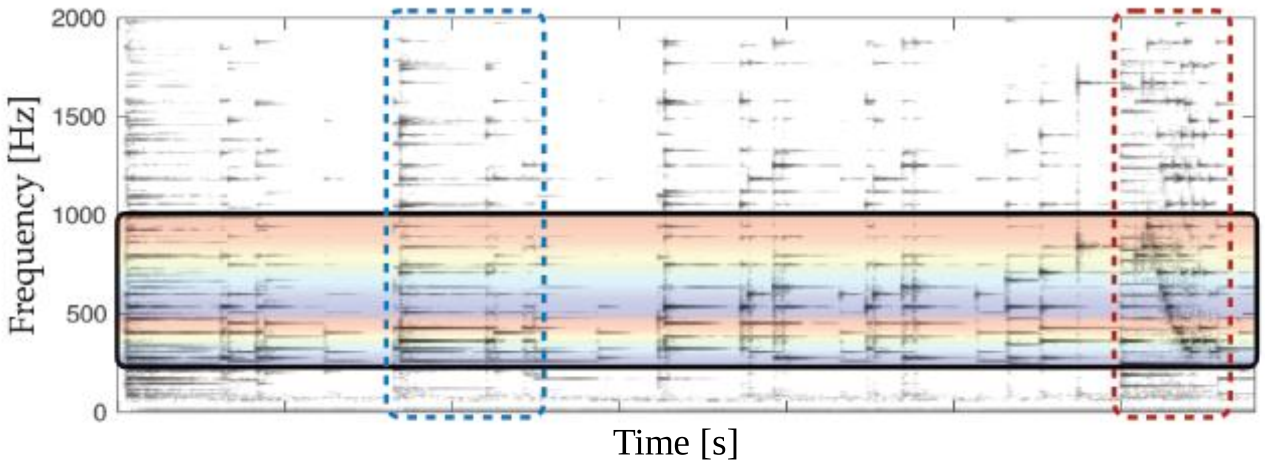
- It is computed by taking the absolute value of the Gabor transform.

- Spectrograms are two-dimensional graphs, with a third dimension represented by colors.
  - **Horizontal axis** represents *Time* that runs from left (oldest) to right (youngest).
  - **Vertical axis** represents *frequency*, with the lowest frequencies at the bottom and the highest at the top.
  - **The third dimension (color/brightness)** represents *amplitude* of a particular frequency at a specific time. Dark reds corresponding to low amplitudes and brighter colors up through blue corresponding to progressively stronger amplitudes.



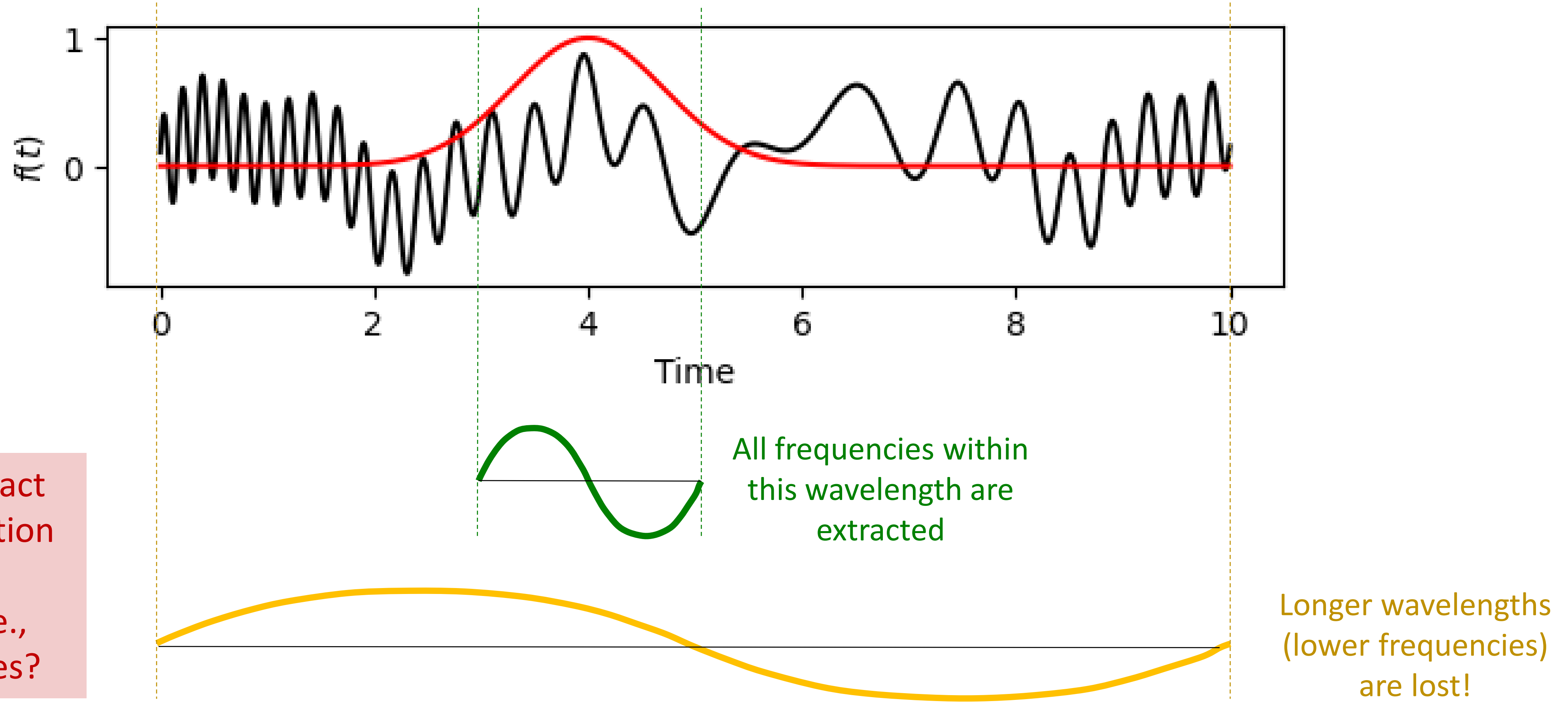
Windowed Fourier Transform

First two bars of Beethoven's Sonata Pathétique



S. Brunton & J. N. Kutz, *"Data-Driven Science And Engineering: Machine Learning, Dynamical Systems"*, and Control, Cambridge University Press, 2019, P73

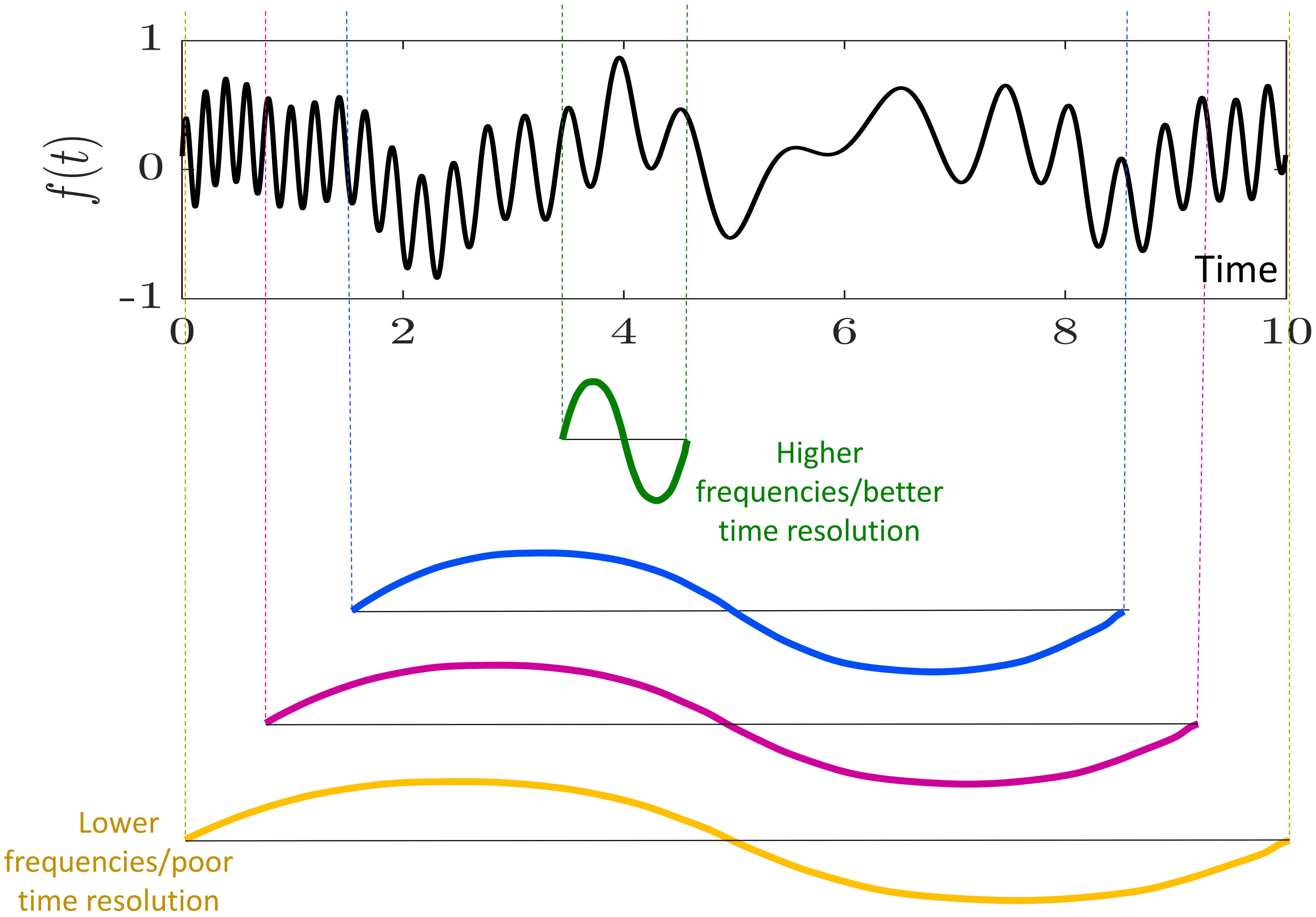
Windowed Fourier Transform



How can we extract the time information from longer wavelengths, i.e., lower frequencies?

Wavelets & Multi-Resolution Analysis

- A simple modification to the Gabor transform allows the scaling window, a, to vary to extract improvements in the time resolution.
- First, the low-frequency (poor time resolution) components are extracted using a broad scaling window.
- The scaling window is subsequently shortened in order to extract higher frequencies and better time resolution.
- By keeping a catalog of the extracting process, both excellent time and frequency resolution of a given signal can be obtained.
- **This is the fundamental principle of wavelet theory.**



Wavelets & Multi-Resolution Analysis

scaling parameter

$$\Psi(a,b) = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{\infty} f(t)\psi^*_{a,b}(t)\mathrm{d}t$$

translation parameter

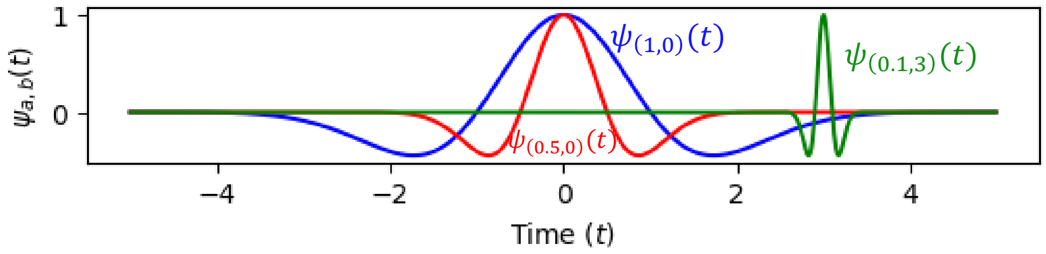
Mother Wavelet

$$\boxed{\psi_{a,b}(t) = \psi\left(\frac{t-b}{a}\right) \quad a \neq 0}$$

Continuous Wavelet Transform.

Mexican Hat Wavelet

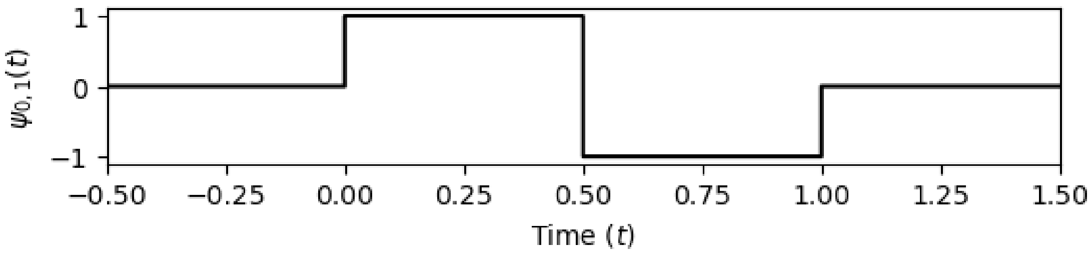
$$\psi_{a,b}(t) = \left[1 - \left(\frac{t-b}{a}\right)^2\right] e^{-\frac{1}{2}\left(\frac{t-b}{a}\right)^2}$$



Discrete Wavelet Transform

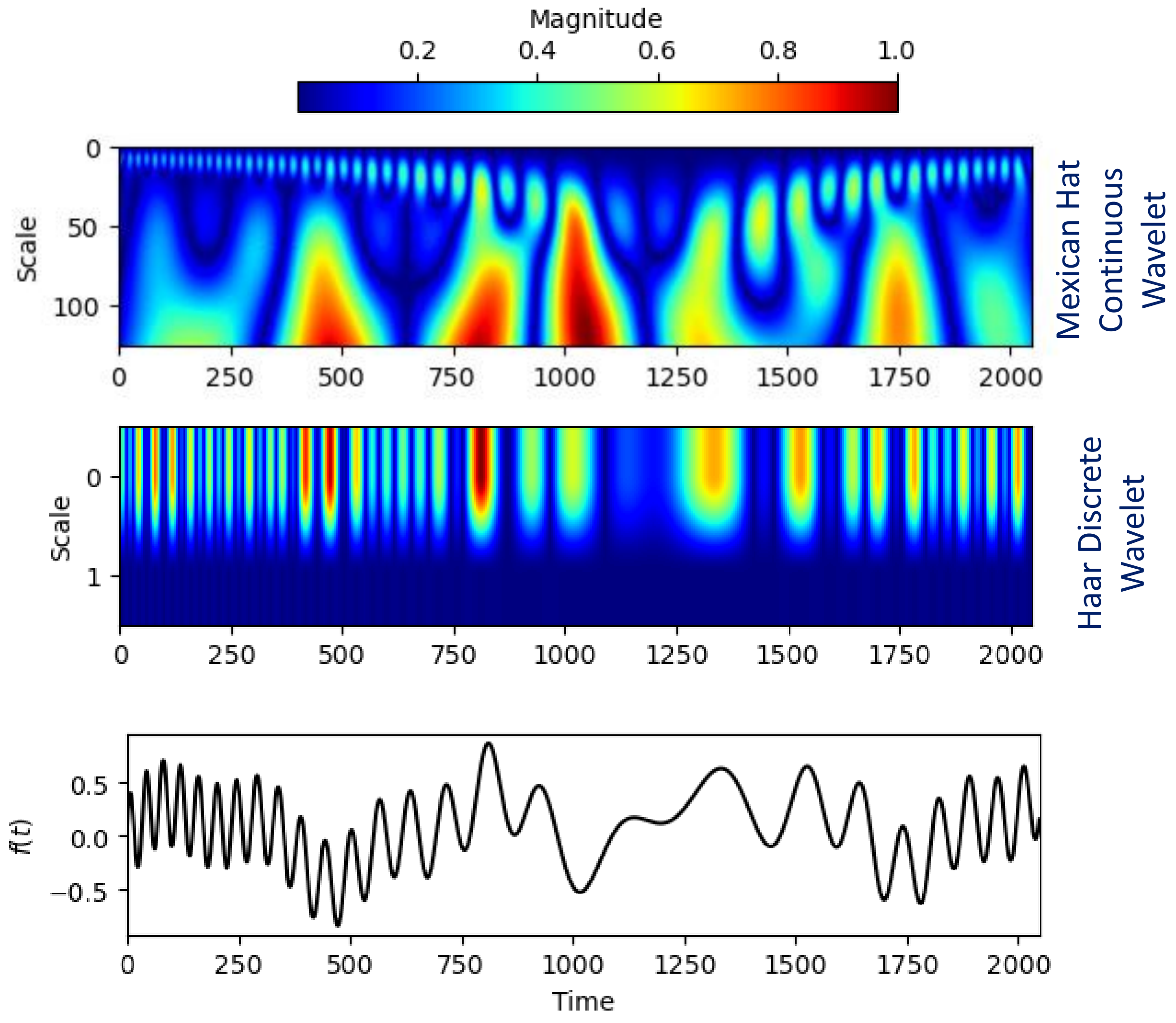
Haar Wavelet

$$\psi_{a,b}(t) = \begin{cases} 1 & \text{for} \quad b \leq t < a/2 \\ -1 & \text{for} \quad a/2 \leq t < a \\ 0 & \text{Otherwise} \end{cases}$$



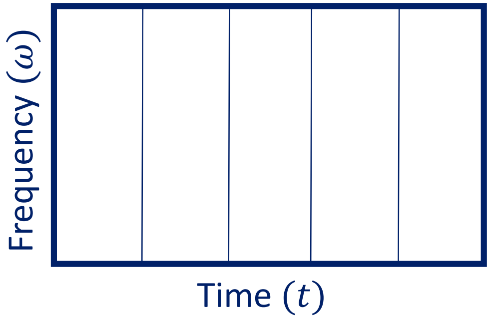
Wavelets & Multi-Resolution Analysis

- A scalogram is a visual representation of a signal's energy distribution (magnitude) over time and scale, generated from the wavelet transform.
- It is computed by taking the absolute value of the wavelet transform.
- Scalograms are two-dimensional graphs, with a third dimension represented by colors.
  - **Horizontal axis** represents *Time* that runs from left (oldest) to right (youngest).
  - **Vertical axis** represents *scale*, which refers to how wide/narrow the wavelet is (wide = low frequency, narrow = high frequency).
  - **The third dimension (color/brightness)** represents *amplitude* (or energy) of the wavelet coefficients.



Graphical Comparison

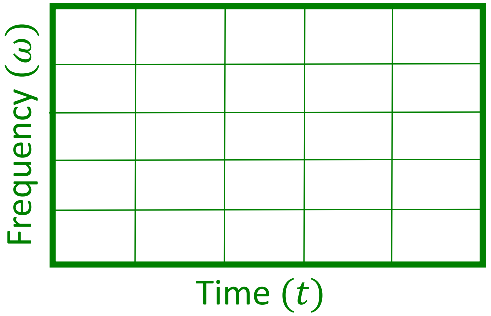
Time-series



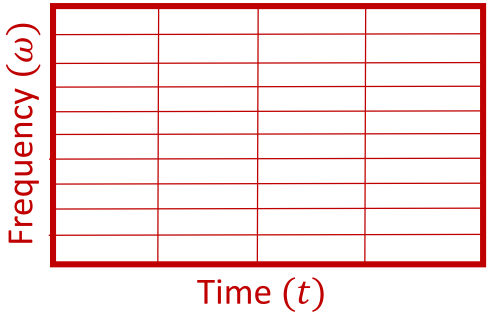
Fourier Transform

Gabor Transform

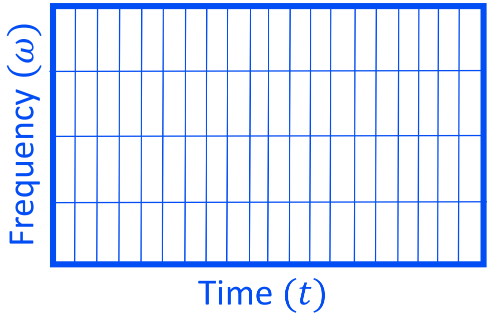
The size of each box refers to the amount of time/frequency Information.



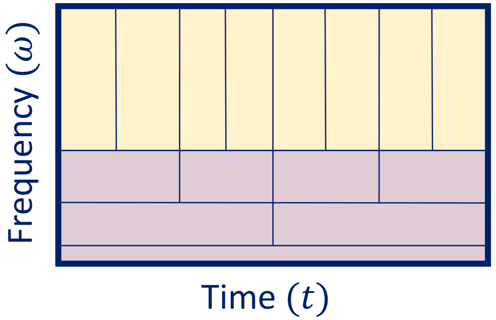
wide windows refer to good accuracy in frequency and less in time



narrow windows refer to a good accuracy in time and less in frequency



Multiresolution Analysis



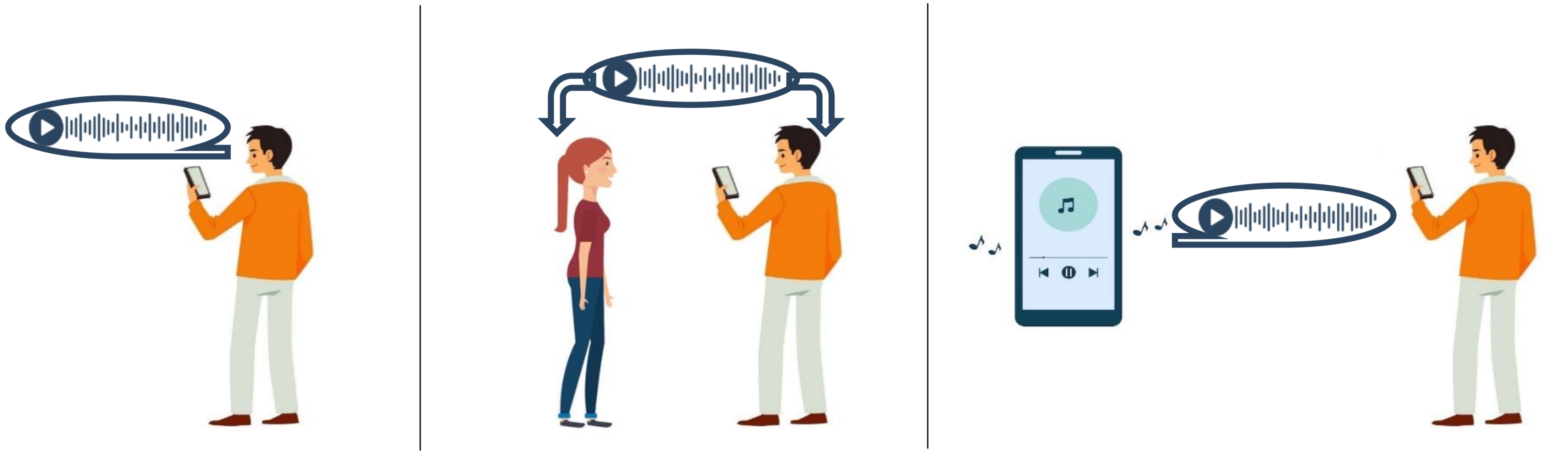
Good time resolution and poor frequency resolution at high frequencies

Good frequency resolution and poor time resolution at low frequencies.

Let’s move on to MATLAB!

Tutorial on Tuesday 03rd Feb

Phyphox: Physical Phone Experiment Software.
Phyphox allows you to use the sensors in your phone for your experiments



(a) (b) (c)

To wrap up with a few reminders…

- **Your third tutorial will be on Tuesday 03rd Feb**. Four PGTAs will be with you during the session. Don't spare any question! Ask them and they will be happy to help.
- All Lecture Material will be uploaded to Moodle later this day, along with the questions of the first tutorial .
- See you next Thursday 05th Feb!