SINTEF

# Aspects of time series analysis

## NorwAI workshop

Alexander Johannes Stasik

August 31, 2023

# Probably everyone knows way more than me

Probably everyone knows way more than me

But maybe there is something new to you as well

# Outline

Introduction

Dynamic time warping

Hilbert transform

Spectrum

Phase coupling

Spectrograms

Wavelets transform

Compressed sensing

Compressed sensing

# What is a time series?

A set of data $\{y(t_1), y(t_2), \ldots, y(t_n)\}$ which is ordered in time such that $t_1 < t_2 < \ldots t_n$ where $y(t_i)$ can depend on all previous data $y(t_j)$ with $t_i > t_j$.

# What is a time series?

A set of data $\{y(t_1), y(t_2), \ldots, y(t_n)\}$ which is ordered in time such that $t_1 < t_2 < \ldots t_n$ where $y(t_i)$ can depend on all previous data $y(t_j)$ with $t_i > t_j$.

Almost all data collected from the real world

# Types of time series

## Continuous time

- Everything related to nature

# Types of time series

## Continuous time

- Everything related to nature
- Weather
- Flow in a pipeline
- Power consumption of a city
- Concentration of a chemical
- ...

# Types of time series

## Continuous time

- Everything related to nature
- Weather
- Flow in a pipeline
- Power consumption of a city
- Concentration of a chemical
- ...

## Discrete time

- Mostly human made

# Types of time series

## Continuous time

- Everything related to nature
- Weather
- Flow in a pipeline
- Power consumption of a city
- Concentration of a chemical
- ...

## Discrete time

- Mostly human made
- Salary statistics (monthly)
- Inflation (monthly)
- Electricity price (daily/hourly)
- 
- ...

Still, we can only measure those systems at discrete time points, so we work with discrete data no matter what

# Generator of time series

We assume there exists a *system* that generates the data.

# Generator of time series

We assume there exists a *system* that generates the data.

**Continuous time**

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = g(x(t), u(t), t)$$

**Discrete time**

$$x_{t+1} = g(x_t, u_t, t)$$

But we typically only observe $y = o(x)$, which is only a snipped of the actual space the system evolves in and we never have the chance to get the full $x$.

# Timescales

Working with time scales can often involve many different time scales. If you take the human heart for example

- Normal heartbeat: order seconds
- Slow evolving heart decease: order month to years
- Heart attach: order minutes /10s of seconds

# Timescales

Working with time scales can often involve many different time scales. If you take the human heart for example

- Normal heartbeat: order seconds
- Slow evolving heart decease: order month to years
- Heart attach: order minutes /10s of seconds

Electric grid

- *Normal* operation 50 Hz
- Slow power line degeneration months/years
- ...

# Domains of time series analysis

Time Resolved Spectrum

Frequency Domain

Time Domain

# Time Series Pre-processing Steps

1. **Handling Missing Values**: Addressing missing data points by imputing or filling with meaningful values.
2. **Smoothing and Filtering**: Applying techniques like moving averages, exponential smoothing, or digital filtering to reduce noise.
3. **Detrending**: Removing trends to focus on underlying patterns.
4. **Normalizing/Scaling**: Scaling data to a common range, e.g., between 0 and 1.
5. **Differencing**: Taking differences between consecutive data points.
6. **Aggregation/Resampling**: Aggregating data at different time intervals or resampling.
7. **Outlier Detection/Handling**: Identifying and handling outlier data points.
8. **Seasonal Decomposition**: Decomposing into trend, seasonal, and residual components.

# Time Series Pre-processing Steps

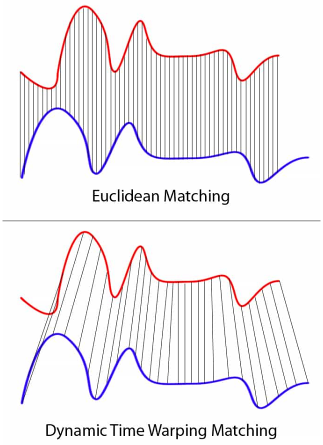
9. **Encoding Timestamps**: Extracting information from timestamps.
10. **Feature Engineering**: Creating new features based on domain knowledge.
11. **Alignment**: Ensuring correct alignment of multiple time series.
12. **Normalization**: Scaling data to mean 0 and standard deviation 1.
13. **Dimensionality Reduction**: Using PCA or SVD to reduce dimensionality.
14. **Handling Seasonality**: Identifying and modeling seasonality patterns.
15. **Dealing with Non-stationarity**: Transforming data to make it stationary.

# Data alignment

Imagine event based data in a time series. Data can be

- Misaligned (shift)
- Squeezed or compressed in time
- Every time index in first sequence get matched with one or more in the second sequence and vice versa
- Obey causality

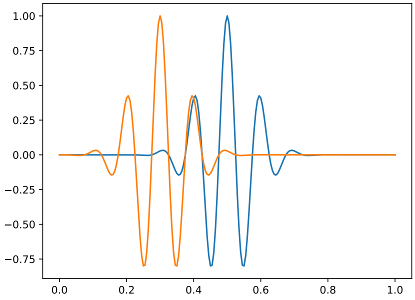
# Dynamic time warping

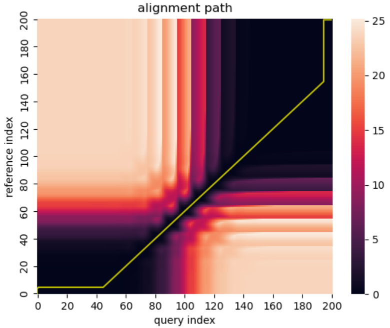


# Dynamic time warping

1. **Alignment Matrix**: DTW creates a distance matirx matrix of aligning point $i$ in the first sequence with point $j$ in the second sequence.
2. **Boundary Constraints**: DTW introduces constraints to ensure that the alignment doesn't start or end too far from the diagonal. These constraints prevent excessive stretching or compressing of the sequences.
3. **Optimal Path**: Starting from the bottom-left cell of the matrix (beginning of both sequences), DTW searches for the path with the lowest cumulative cost while staying within the boundary constraints. The path typically moves diagonally, but can also move to the right or up.
4. **Accumulated Distance**: The accumulated distance along the optimal path represents the similarity between the two sequences, accounting for local variations in timing and speed.

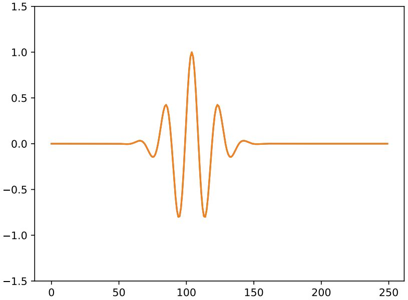
# Dynamic time warping



# Dynamic time warping

- DTW one of many methods for alignment
- Can be computationally heavy
- Relatively robust to noise
- Can be used on other types of distance to align



# Hilbert transform and analytic signal

Characterization of an oscillatory system $x(t) = \sum_i A_i(t) \sin(\omega_i(t)t + \phi_i(t))$

- (Current frequency)
- Amplitude
- Phase (difference)

# Hilbert transform and analytic signal

Characterization of an oscillatory system $x(t) = \sum_i A_i(t) \sin(\omega_i(t)t + \phi_i(t))$

- (Current frequency)
- Amplitude
- Phase (difference)

## Hilbert transform

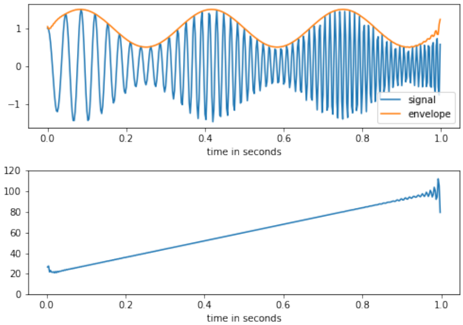
$$\mathcal{H}(u)(t) = \frac{1}{\pi} \mathrm{p.v.} \int_{-\infty}^{+\infty} \frac{u(\tau)}{t-\tau} \mathrm{d}\tau$$

returns the *analytic signal*.



SINTEF

# Hilbert transform and analytic signal




- Real part of $\mathcal{H}(u)(t)$ is the original signal
- Norm of $\mathcal{H}(u)(t)$ is the envelope/amplitude
- Angle in polar coordinates of $H(u)(t)$ gives the instantaneous phase
- One can also define the instantaneous frequency

# Fourier transformation

Theory reminder

## Fourier transform

Any period function can be represented in the basis of complex harmonics $\mathrm{e}^{-\mathrm{i}\omega t}$ as

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)\mathrm{e}^{-\mathrm{i}\omega t}\mathrm{d}t$$

where $F(\omega)$ is the strength at frequency $\omega$. $F(\omega) \in \mathbb{C}$ where the real part is the amplitude and the complex part is the phase.

Practically, one does this at discrete values, leading to the discrete Fourier transform.

# Fourier transformation

## Fourier transform

The discrete version is given by

$$X_k = \sum_{n=0}^{N-1} x_n \mathrm{e}^{-\frac{2\pi \mathrm{i}}{N} kn}$$

transforming from a sequence $\{x_0, x_1, \ldots, x_{n-1}\}$ to $\{X_0, X_1, \ldots, X_{n-1}\}$

# Fourier transformation

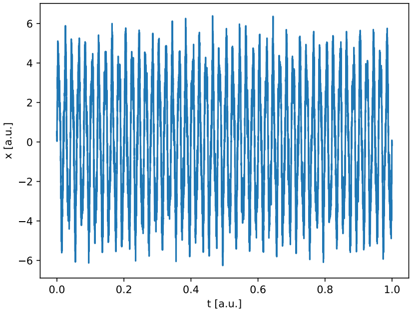
- Very cool mathematical tool
- Can be used on any function
- Super helpful for proofs, etc

# Fourier transformation

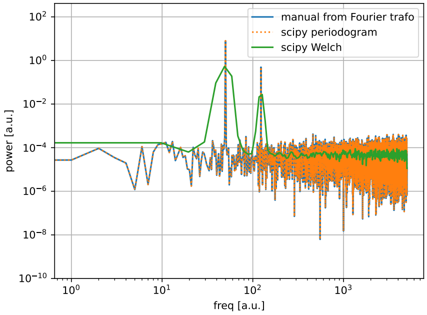
- Very cool mathematical tool
- Can be used on any function
- Super helpful for proofs, etc

- In practice, mostly used on oscillatory data
- Use a discrete version on discrete data, never solve the integral
- Mostly used for power spectra ($F(\omega)^2$), phase often ignored
- In vanilla version, does not provide any temporal resolution
- More advanced methods, more stable to noise, overlapping windows

# Spectrum



6
4
2
0
−2
−4
−6
x [a.u.]
0.0
0.2
0.4
0.6
0.8
1.0
t [a.u.]

manual from Fourier trafo
scipy periodogram
scipy Welch
$10^{2}$
$10^{0}$
$10^{-2}$
$10^{-4}$
$10^{-6}$
$10^{-8}$
$10^{-10}$
power [a.u.]
$10^{0}$
$10^{1}$
$10^{2}$
$10^{3}$
freq [a.u.]

# Phase coupling

Given a time series, we are interested if two frequencies have a fixed phase

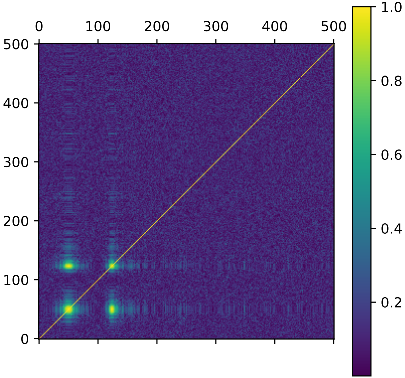
- Some measure of *causality*
- Pure noise is expected to have a random phase
- If e.g. something is driven in the system, driver and oscillator will have a fixed phase relation
- Multiple ways of defining it

# Phase coupling

**One definition**

$$b(f_1, f_2) = \frac{\left|\sum_i^N F_i(f_1) F_i^*(f_2)\right|}{\sum_i^N \left|F_i(f_1) F_i^*(f_2)\right|}$$

with $F_i(\cdot)$ as Fourier transform. The time series is split into $N$ snippets (overlap is allowed).





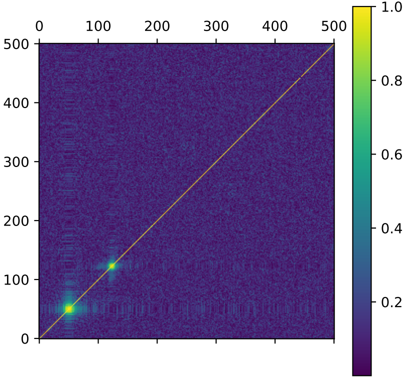

SINTEF

# Phase coupling

**One definition**

$$b(f_1, f_2) = \frac{\left| \sum_i^N F_i(f_1) F_i^*(f_2) \right|}{\sum_i^N \left| F_i(f_1) F_i^*(f_2) \right|}$$

with $F_i(\cdot)$ as Fourier transform. The time series is split into $N$ snippets (overlap is allowed).






SINTEF

# Intuition

$$
\begin{aligned}
F(f_1)F^*(f_2) &= A_1 e^{-i\phi_1} A_2 e^{+i\phi_2} \\
&= A_1 A_2 e^{-i(\phi_1 - \phi_2)}
\end{aligned}
$$

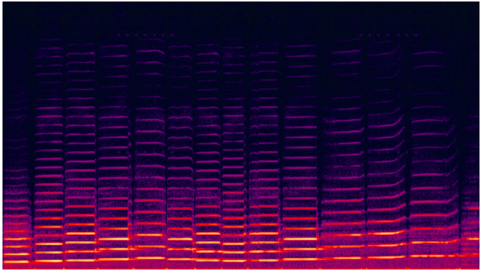
If we focus on the term $e^{-i(\phi_1 - \phi_2)}$ and sum many of those with random phases, so

$$
\sum_i e^{-i(\phi_1^i - \phi_2^i)}
$$

with $\phi \sim \mathcal{U}(0, 2\pi)$, eventually, it will all cancel out. If both values are fixed thought (or their difference), there is no canceling.

# Spectrogram

- The spectrum assumes stationary signal
- All information about time is lost in the process
- Spectrograms are a way to have both





SINTEF

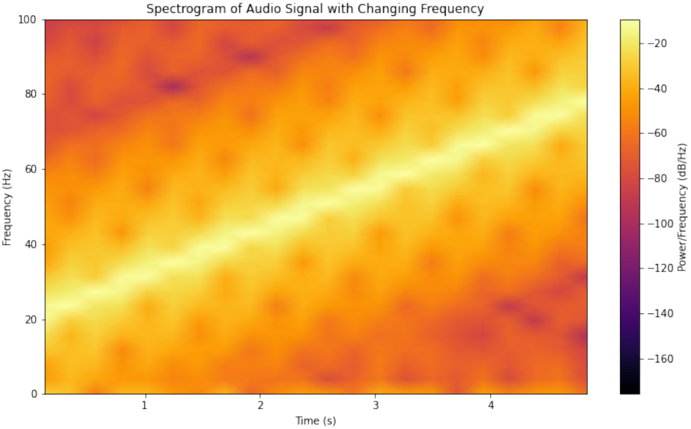
# Spectrograms

- Cut the time series in (overlapping) windows
- Compute a spectrum for each slice
- Works well if one has dense data and windows are very large

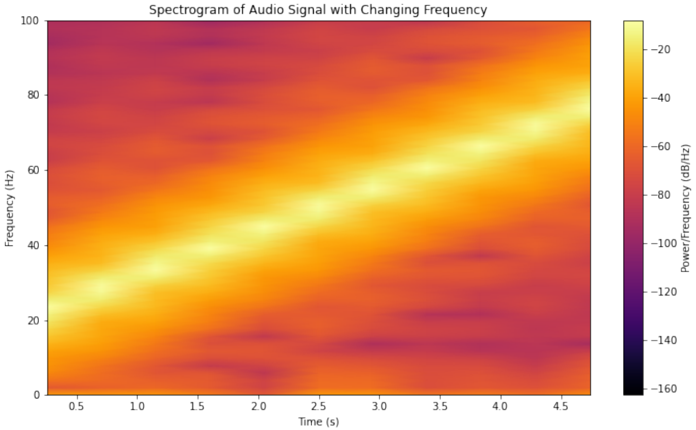


SINTEF

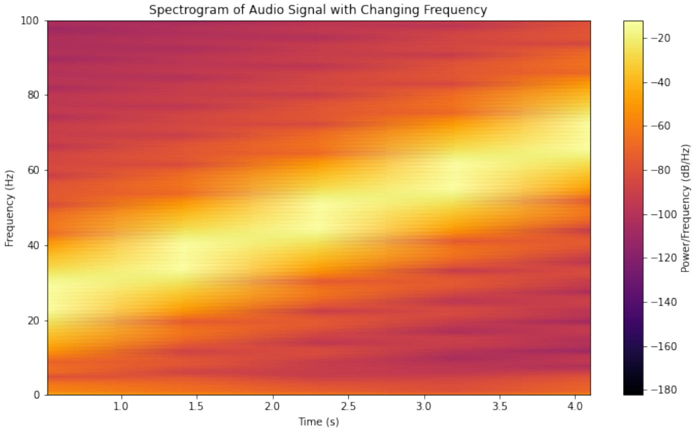
# Spectrograms



# Spectrograms



# Spectrograms






SINTEF

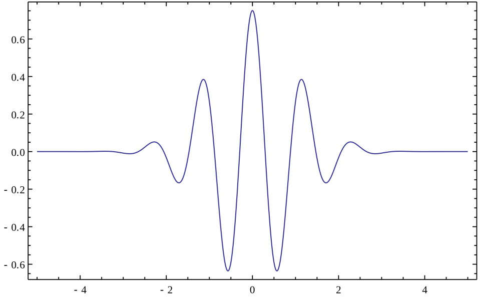
# Spectrograms

- Cut the time series in (overlapping) windows
- Compute a spectrum for each slice
- Works well if one has dense data and windows are very large
- High frequencies require shorter duration than lower frequencies
- We want typically high resolution at high frequencies
- Trade off, either lose low frequencies or temporal resolution at high frequencies

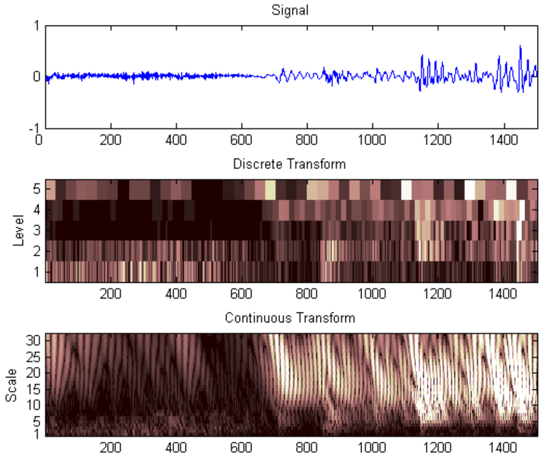
# Wavelets

Idea: Use a different window for every frequency

- Wavelets are localized functions
- They are convoluted over the time signal and give a large magnitude where they *match* best
- They are then translated and scaled to match different frequencies
- Mathematically, they form a basis of Hilbert space, so they are arbitrary close to any signal
- There are many different wavelets, performance depend on the choice



# Wavelets



Signal
1
0
-1
0
200
400
600
800
1000
1200
1400
Discrete Transform
Level
5
4
3
2
1
200
400
600
800
1000
1200
1400
Continuous Transform
Scale
30
25
20
15
10
5
1
200
400
600
800
1000
1200
1400

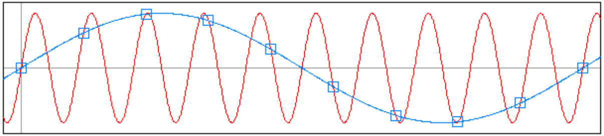
# Wavelets

- Mathematical, wavelet analysis is solid
- Practically, choice of wavelet is critical
- Wavelets are complex functions and are extended into $\mathbb{C}^n$, so can be used for images, etc
- Interpretation of frequency and power is a bit tricky

# Compressed sensing - can we beat the Nyquist theorem?

## Nyquist theorem

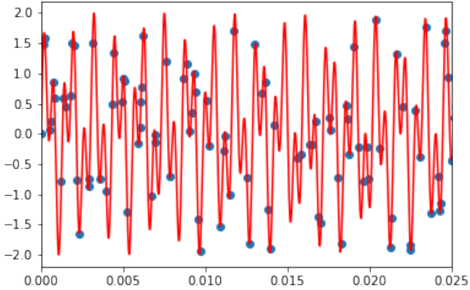
Theorem — If a function $x(t)$ contains no frequencies higher than $B$ hertz, then it can be completely determined from its ordinates at a sequence of points spaced less than $1/(2B)$ seconds apart.



# Compressed sensing - can we beat the Nyquist theorem?

Observation: Natural signals are typically sparse

1. Randomly sample from a distribution
2. Compute a *sparse* Fourier transformation ($|\cdot|_1$) regularization
3. Transform back into time domain



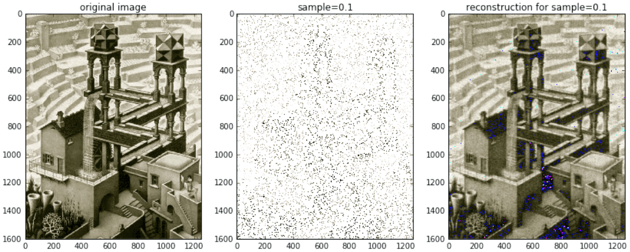



SINTEF

# Compressed sensing - can we beat the Nyquist theorem?

Observation: Natural signals are typically sparse

1. Randomly sample from a distribution
2. Compute a *sparse* Fourier transformation ($|\cdot|_1$) regularization
3. Transform back into time domain






SINTEF

# Whats next

- Lunch

# Whats next

- Lunch
- Small examples for all those methods
- Feel free to play with it
- Invitation for discussion