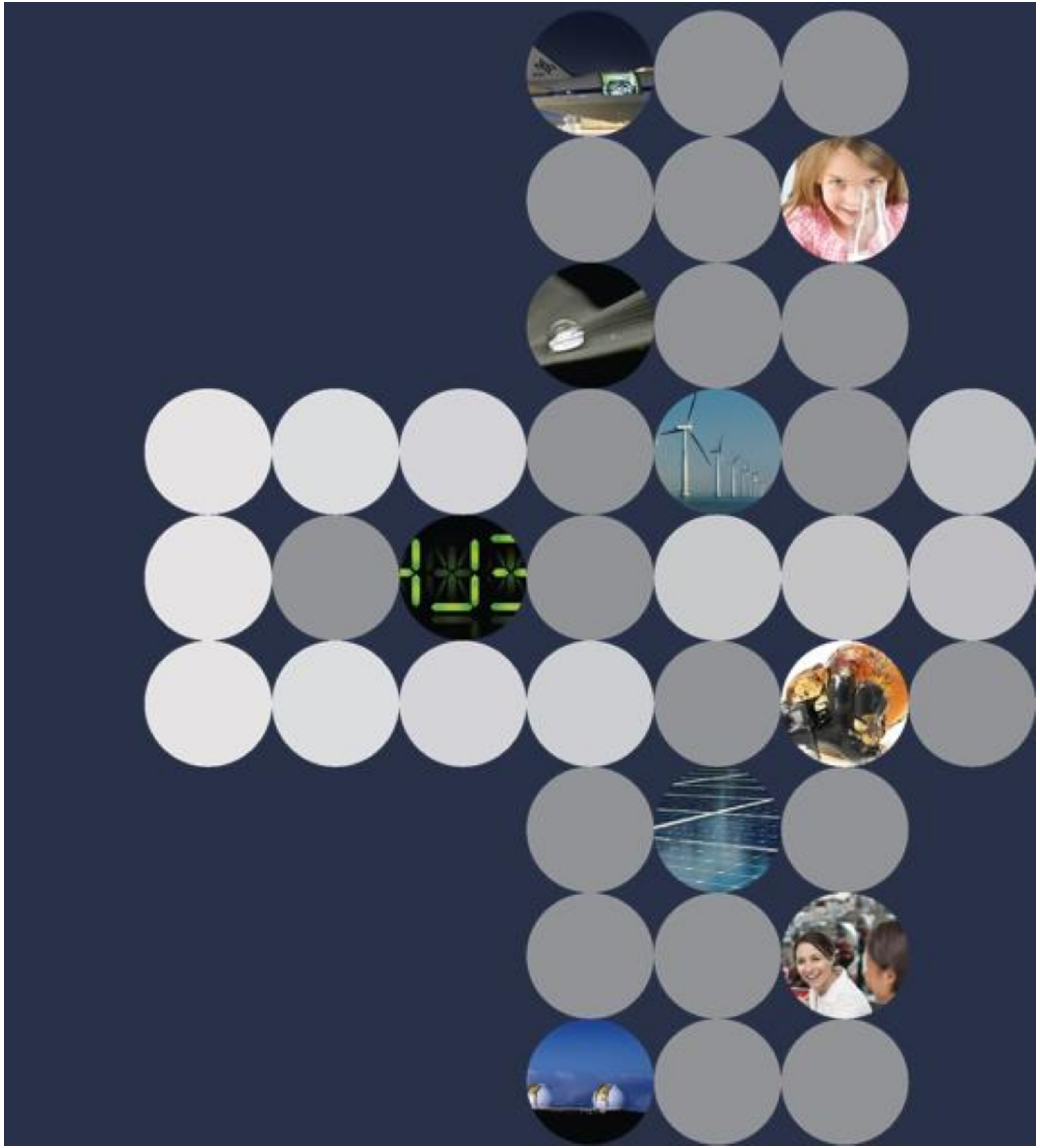
EE512 – Applied Biomedical Signal Processing

Time & Frequency

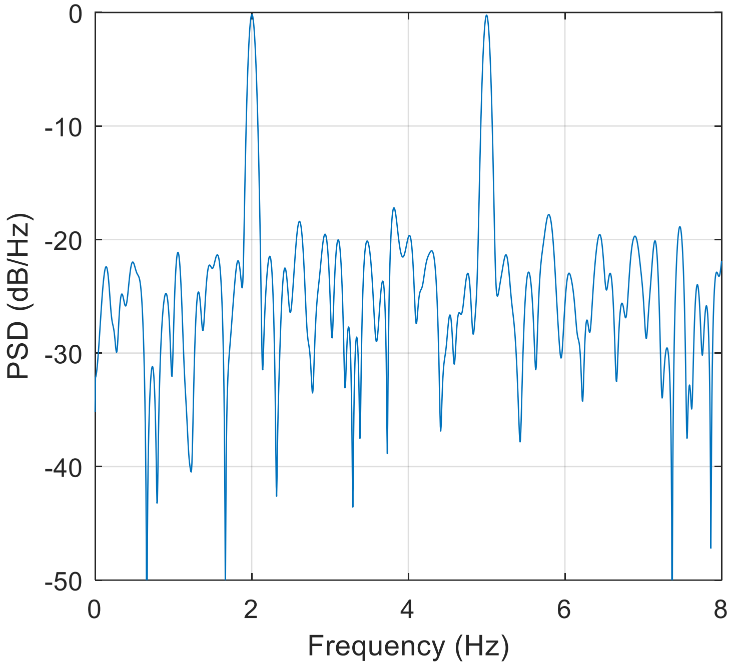
Martin PROENÇA
CSEM Signal Processing Group



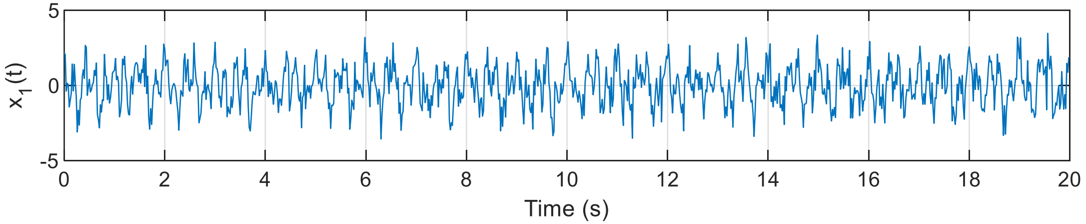
Outline

- Motivation for time-frequency analysis
- The short-time Fourier transform (STFT)
- Spectrograms in practice
- Examples
- Alternative to the STFT: Wavelet analysis

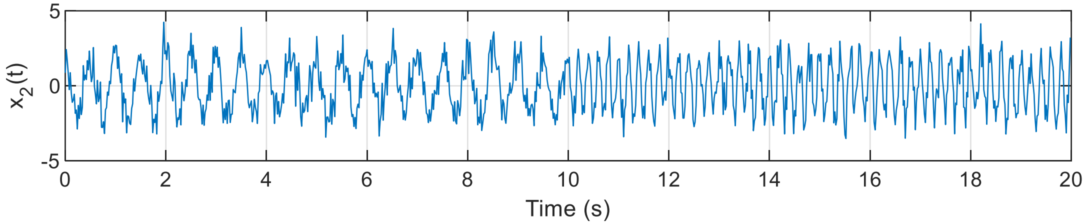
Motivation for time-frequency analysis



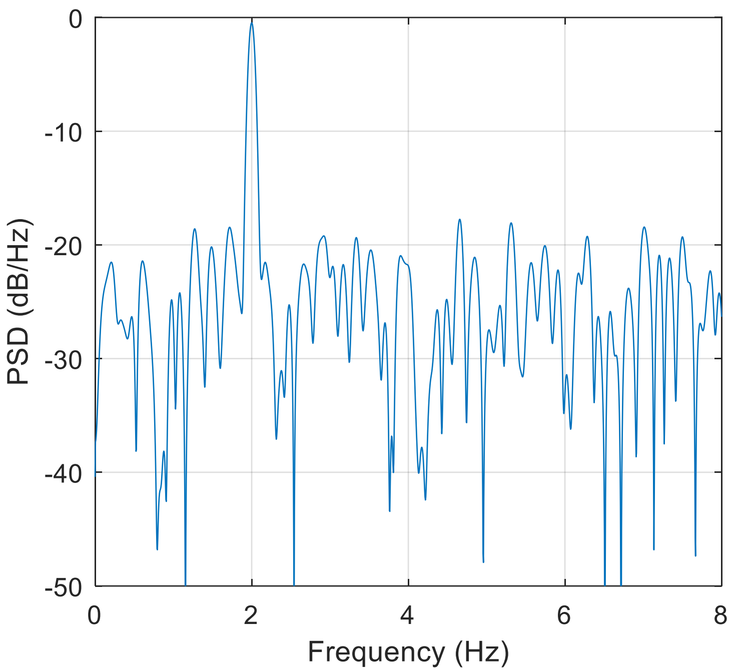
$$x_1(t) = \cos(2\pi \cdot 2t) + \cos(2\pi \cdot 5t) + \text{noise}$$



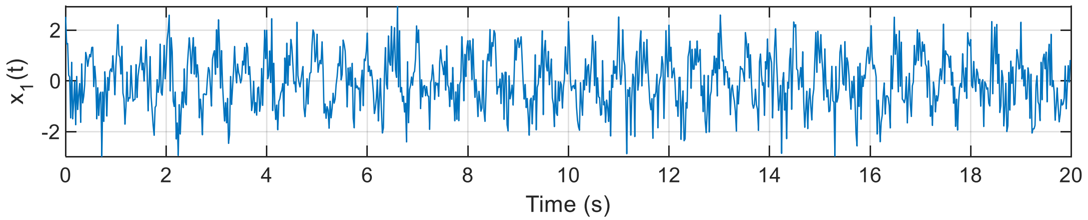
$$x_2(t) = \begin{cases} 2\cos(2\pi \cdot 2t) & 0s \leq t < 10s \\ 2\cos(2\pi \cdot 5t) & 10s \leq t < 20s \end{cases} + \text{noise}$$



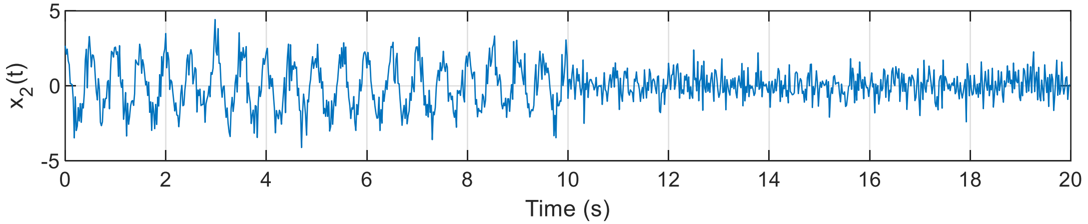
Motivation for time-frequency analysis



$$x_1(t) = \cos(2\pi \cdot 2t) + \text{noise}$$



$$x_2(t) = \begin{cases} 2\cos(2\pi \cdot 2t) & 0s \leq t < 10s \\ 0 & 10s \leq t < 20s \end{cases} + \text{noise}$$



EPFL csem

Motivation for time-frequency analysis

- The **Fourier transform** was developed for **stationary** signals, i.e. for signals whose properties (frequency, amplitude, etc.) are assumed **not** to change over time

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

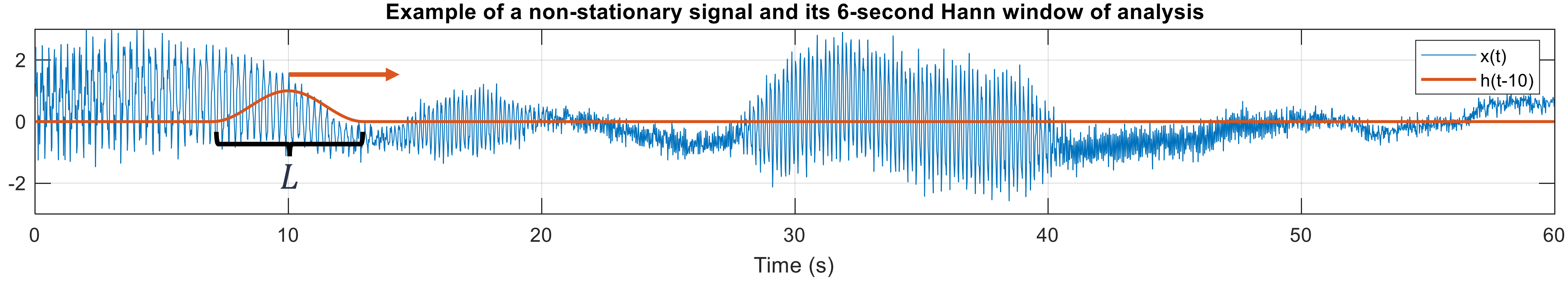
- When the signal is **non-stationary**, **Fourier analysis is limited** and can produce a very incomplete and confusing picture of the signal
- Intuitive solution: Split the signal into short **segments in which the signal can be assumed to be stationary** and perform Fourier analysis on each segment. This is the **short-time Fourier transform (STFT)**.

Short-term Fourier Transform (STFT)

- The STFT of a signal $x(t)$ is given by:

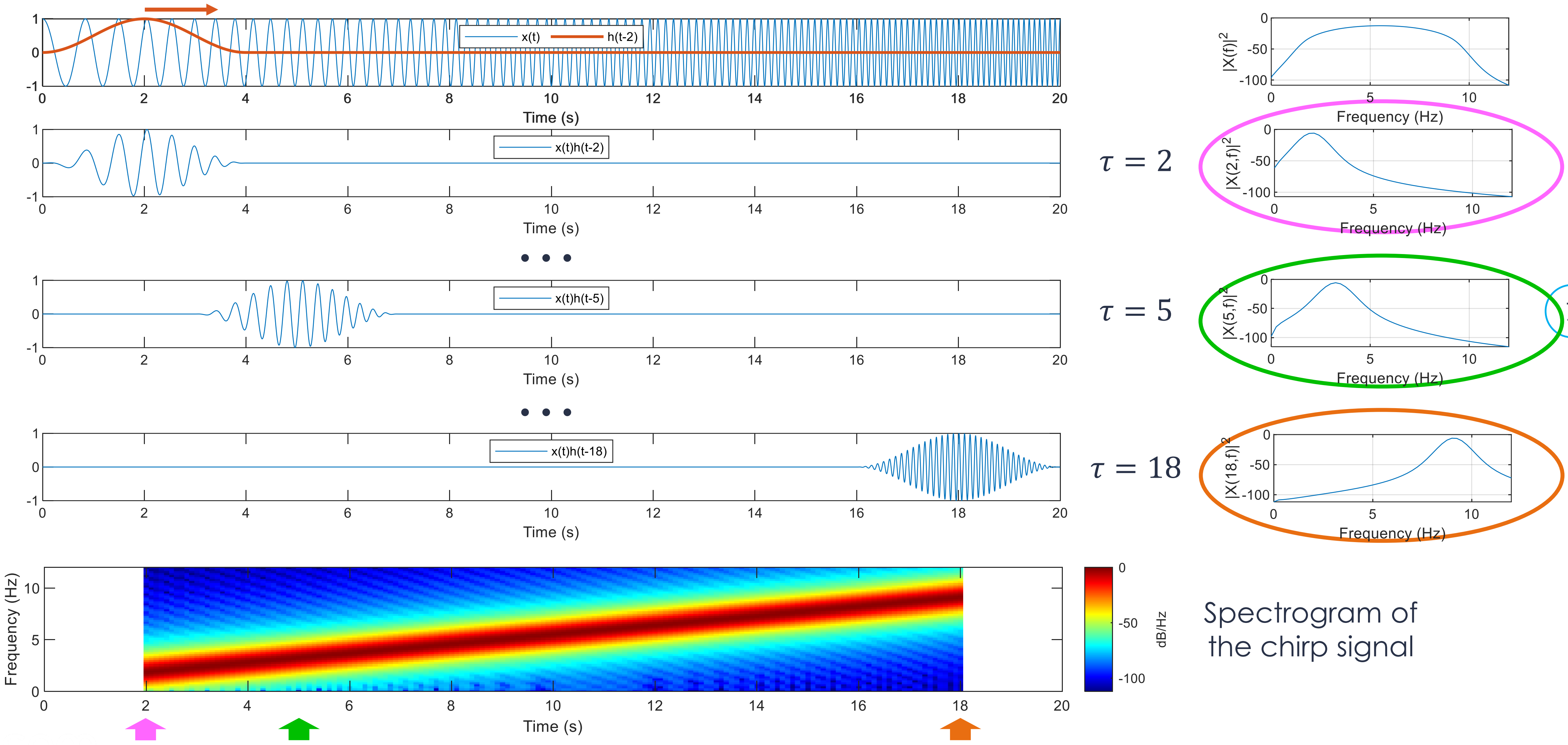
$$X_h(\tau, f) = \int_{-\infty}^{\infty} x(t)h(t-\tau)e^{-i2\pi f t}dt$$

where $h(t-\tau)$ is a window function centered around $t = \tau$. The window of length L is zero everywhere outside the interval $\tau \pm L/2$.



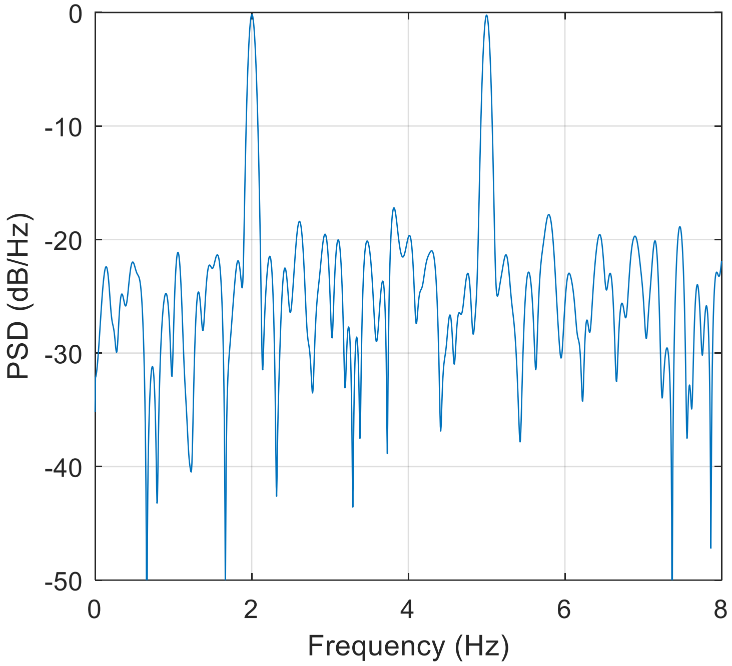
EPFL csem

Short-term Fourier Transform (STFT): Example on a chirp signal

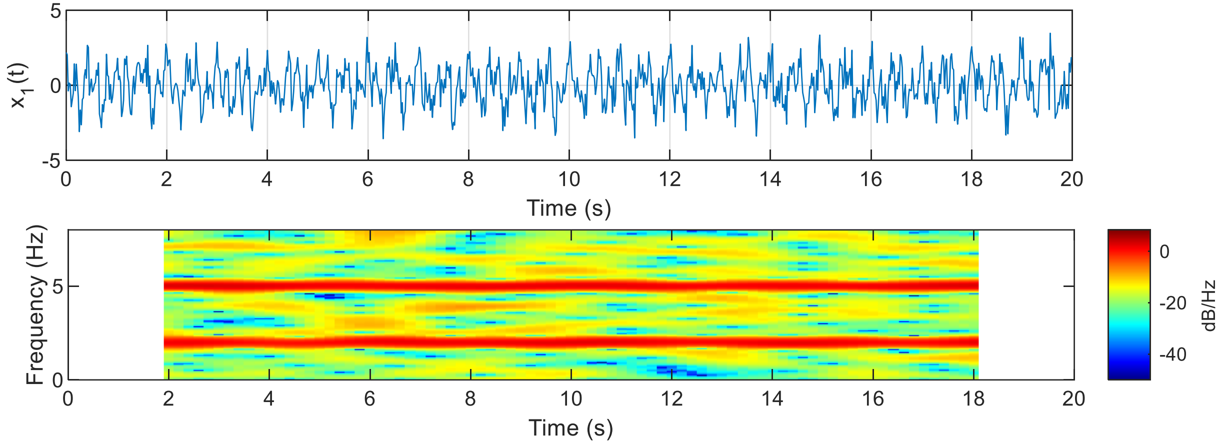


Spectrogram of the chirp signal

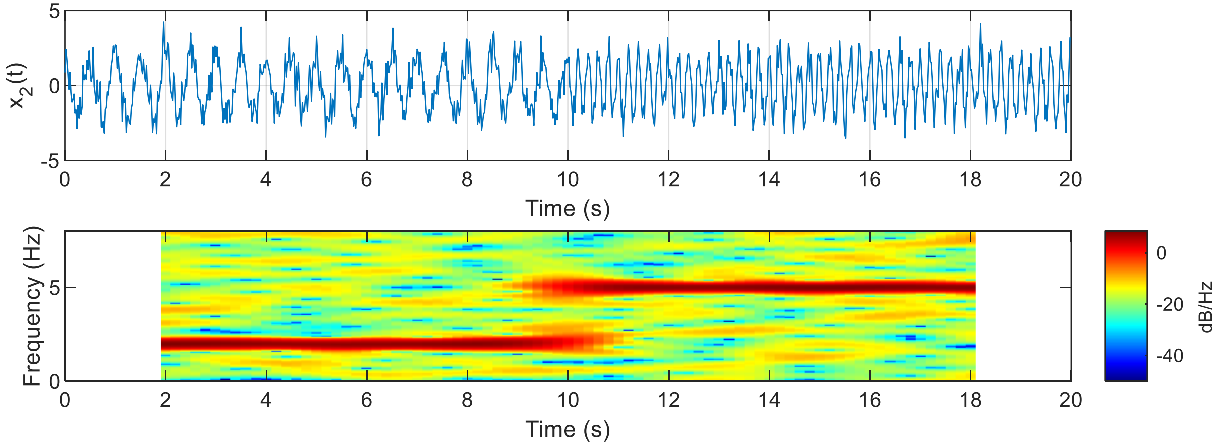
Motivation for time-frequency analysis



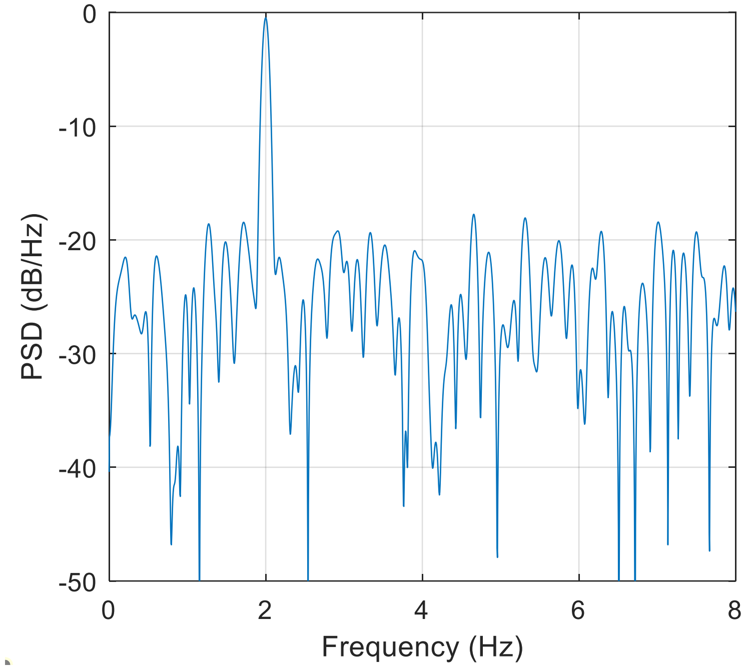
$$x_1(t) = \cos(2\pi \cdot 2t) + \cos(2\pi \cdot 5t) + \text{noise}$$



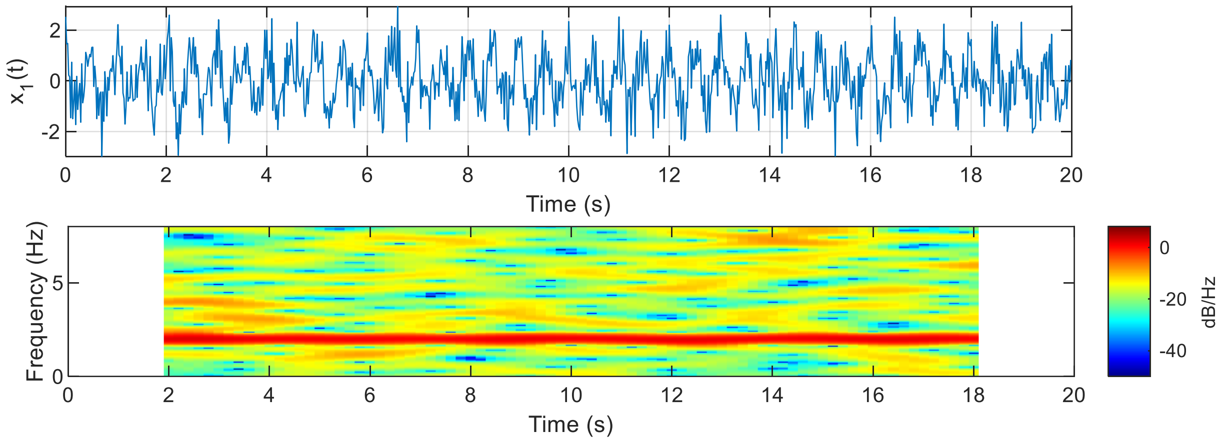
$$x_2(t) = \begin{cases} 2\cos(2\pi \cdot 2t) & 0s \leq t < 10s \\ 2\cos(2\pi \cdot 5t) & 10s \leq t < 20s \end{cases} + \text{noise}$$



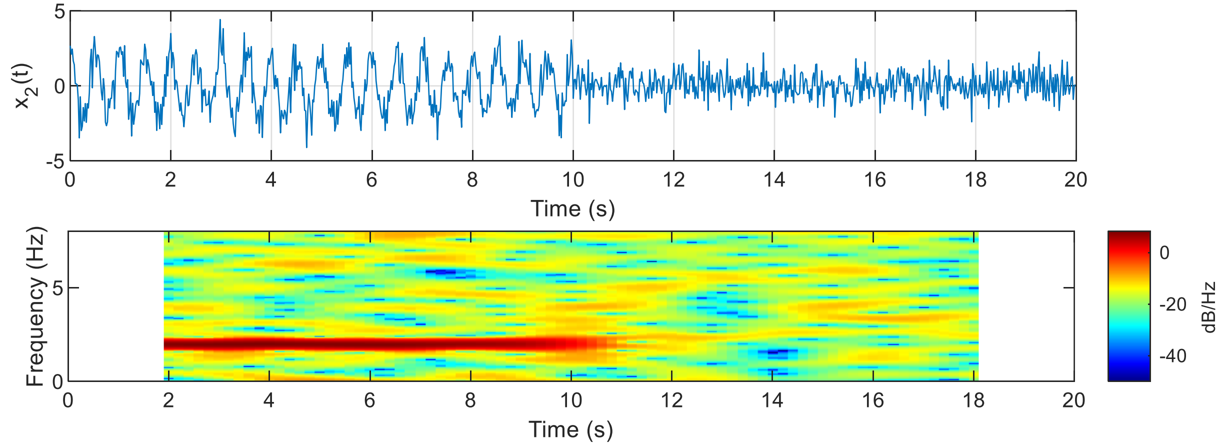
Motivation for time-frequency analysis



$$x_1(t) = \cos(2\pi \cdot 2t) + \text{noise}$$



$$x_2(t) = \begin{cases} 2\cos(2\pi \cdot 2t) & 0s \le t < 10s \\ 0 & 10s \le t < 20s \end{cases} + \text{noise}$$



EPFL csem

Spectrograms in practice: Main parameters

- Window length
- Window function
- Overlap
- Number of FFT points

- In Matlab: `spectrogram(x, window, noverlap, nfft, fs)`

Window length, Window function → `window`; Overlap → `noverlap`; Number of FFT points → `nfft`

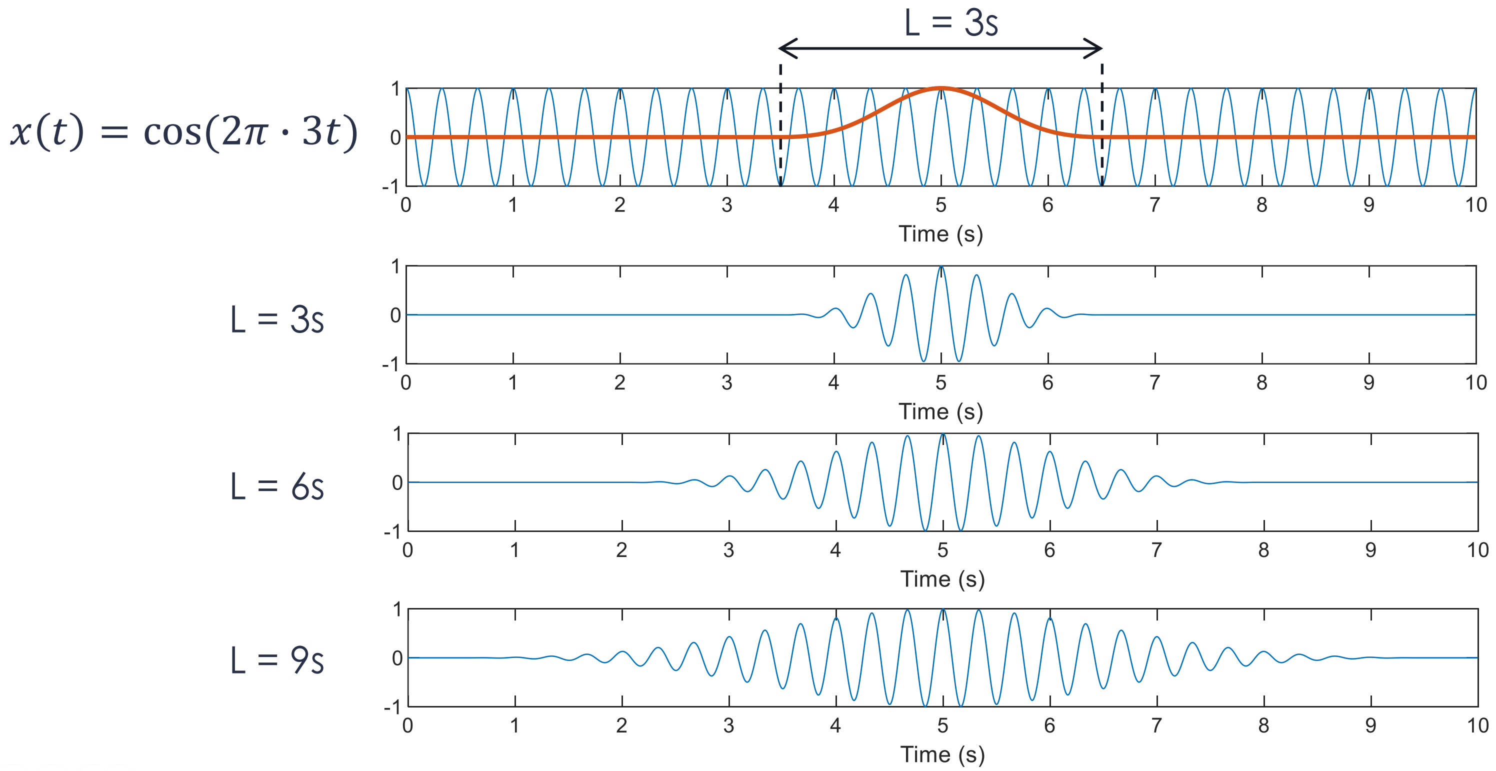
Signal → `x`

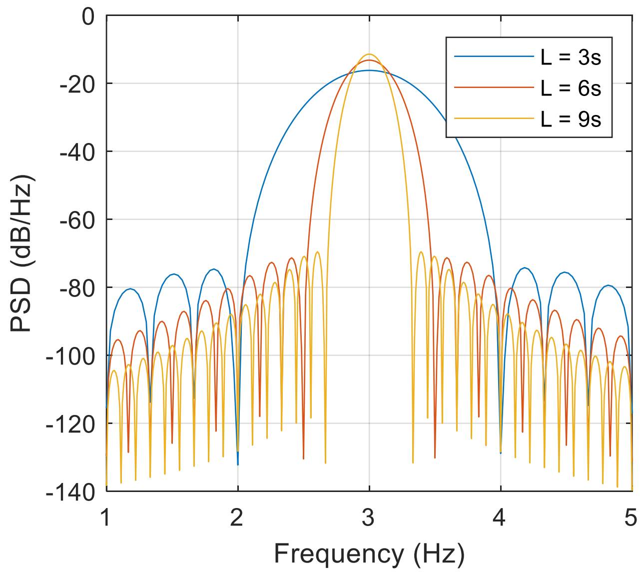
Sampling frequency → `fs`

Spectrograms in practice: Main parameters
Window length

- Longer → Better frequency resolution & Lower time resolution
- Shorter → Lower frequency resolution & Better time resolution

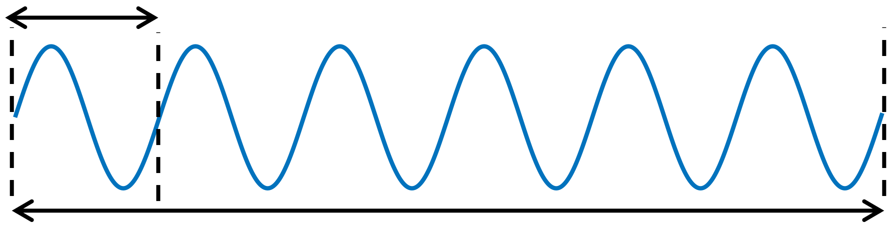
} Uncertainty principle → Trade-off required





Spectrograms in practice: Main parameters
Window length

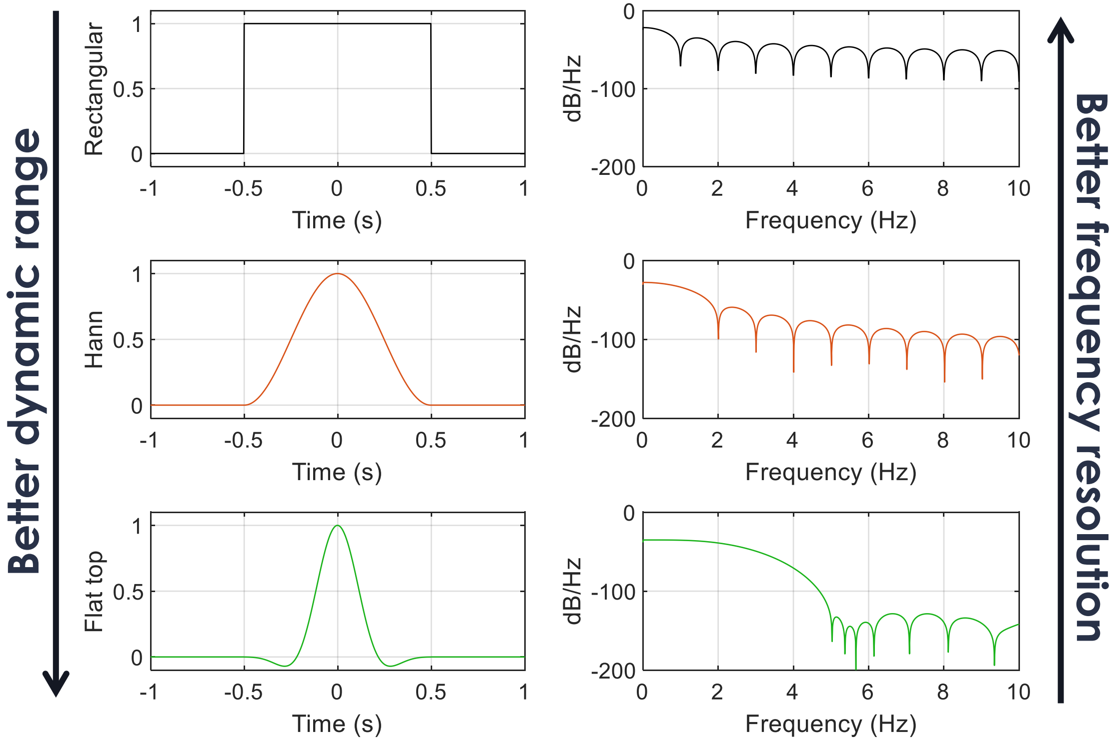
- In theory, window length must be **short enough for the stationarity assumption to always be true.** In practice, guaranteeing stationarity at all times is rarely possible, particularly in transitory parts of the signal.
- In theory, window length must be **long enough to include at least one period of the lowest frequency** in the band of interest. In practice, it is often preferrable to chose windows long enough to include at least a few periods of the lowest frequency for improved frequency resolution.

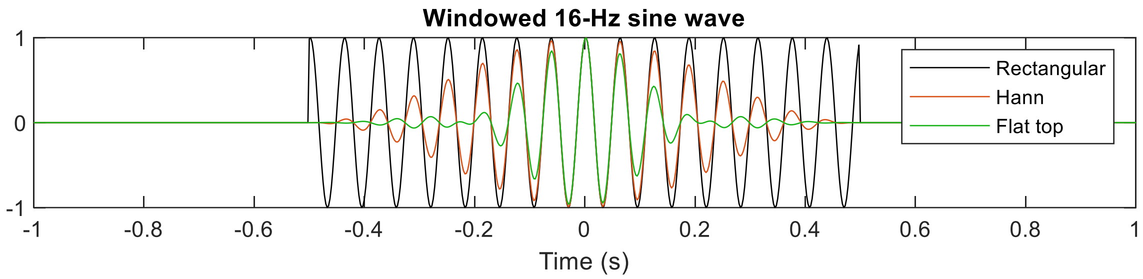


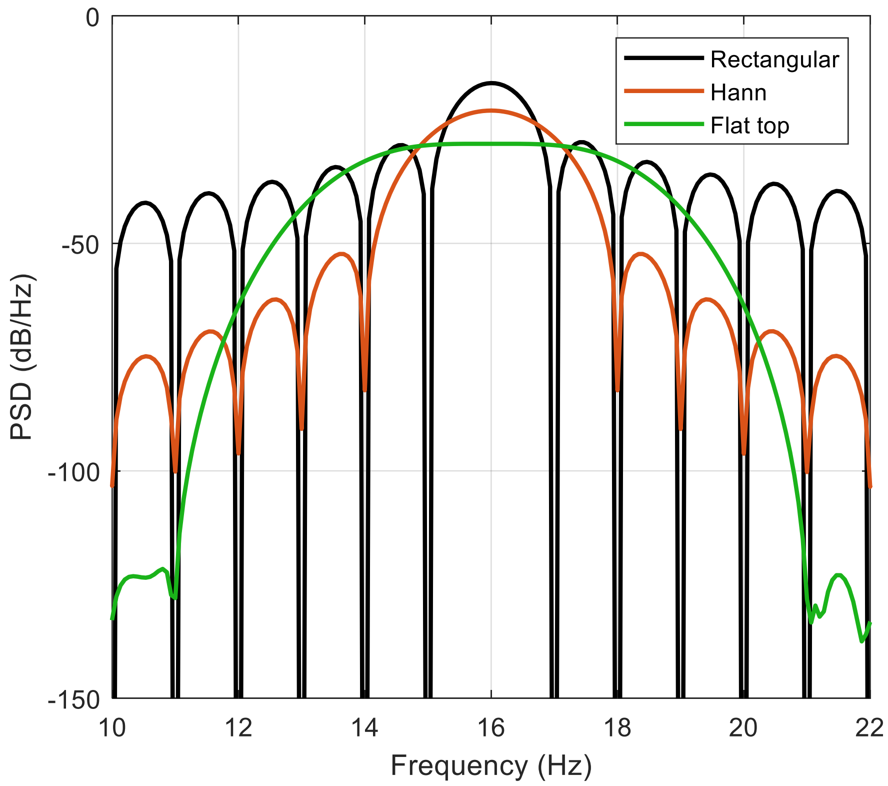
- If there are still several possible choices for the window length after that, the **frequency vs. time resolution** considerations presented in the last slide come into play.

Spectrograms in practice: Main parameters

Window function



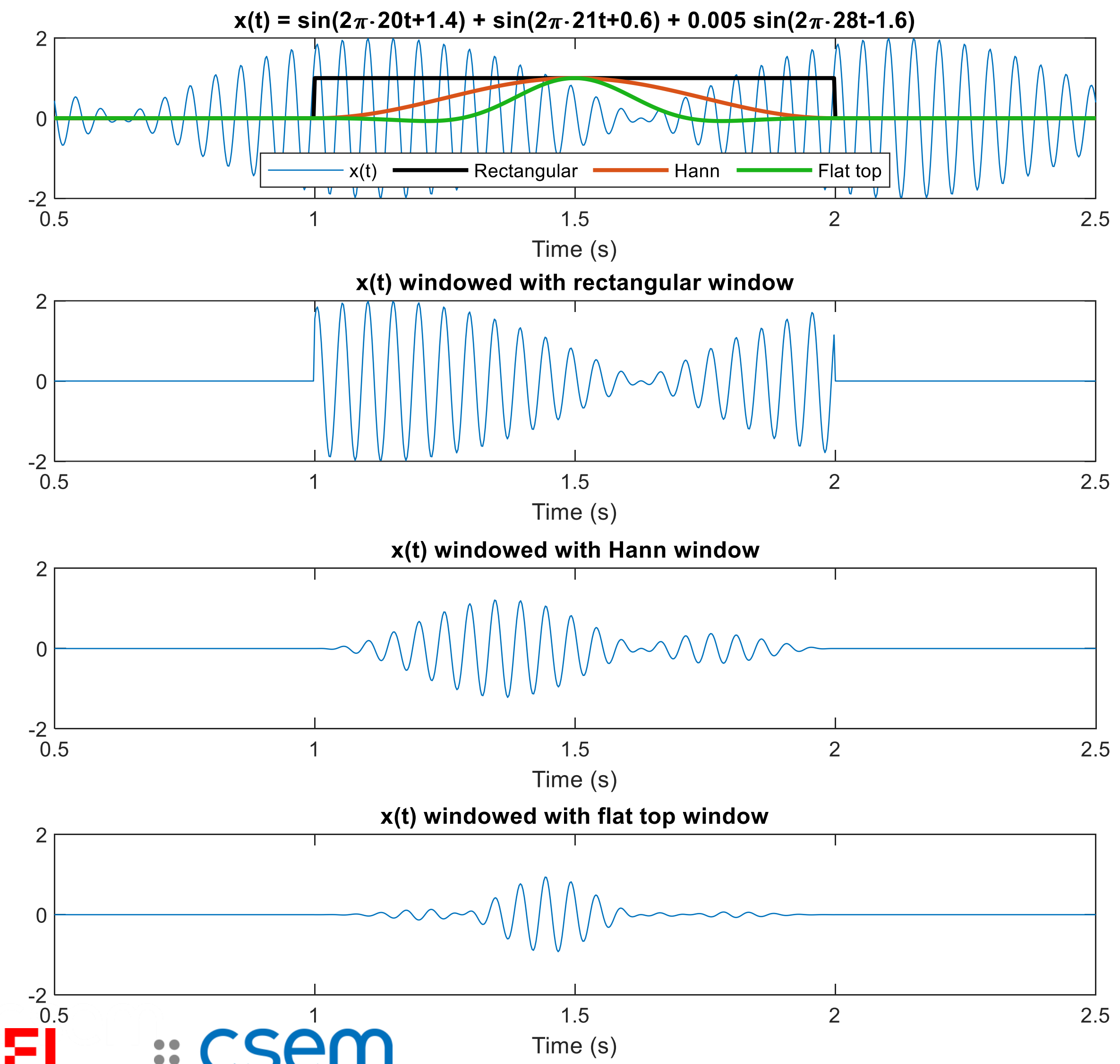


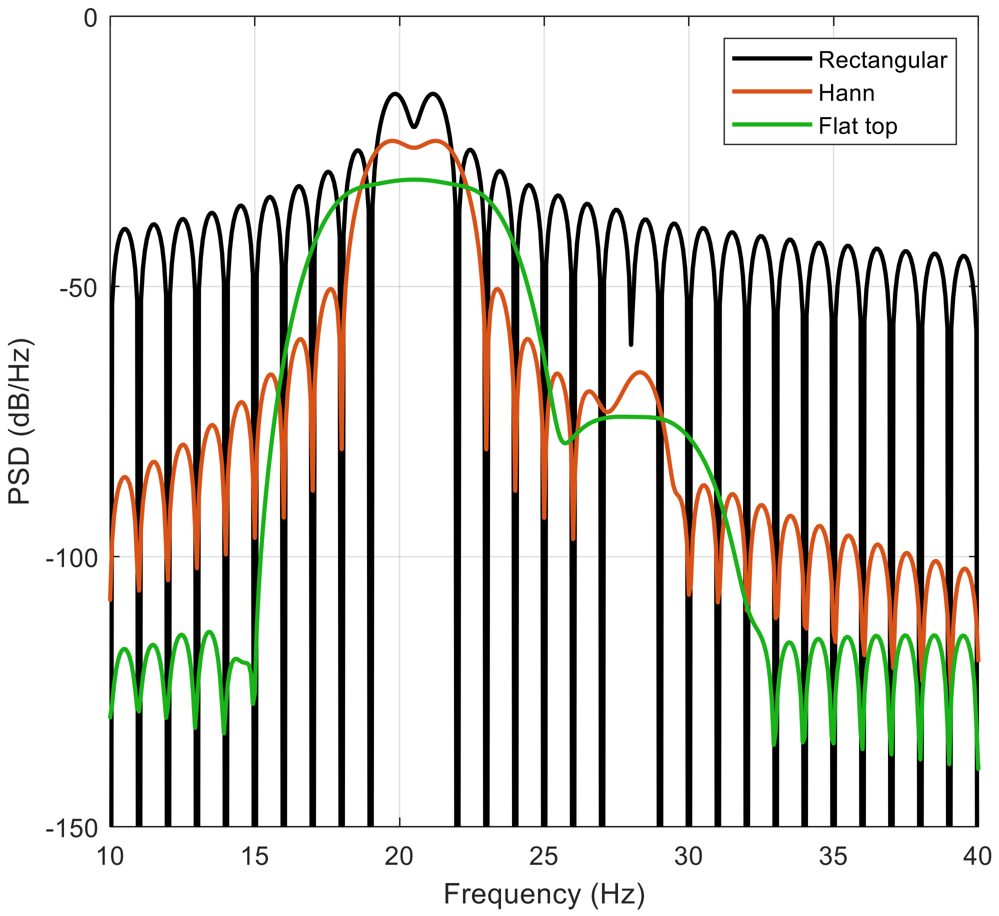


- ➢ **Frequency resolution:** ability to distinguish frequencies that are close to each other
- ➢ **Dynamic range:** ability to distinguish frequencies of different strengths

Spectrograms in practice: Main parameters

Window function





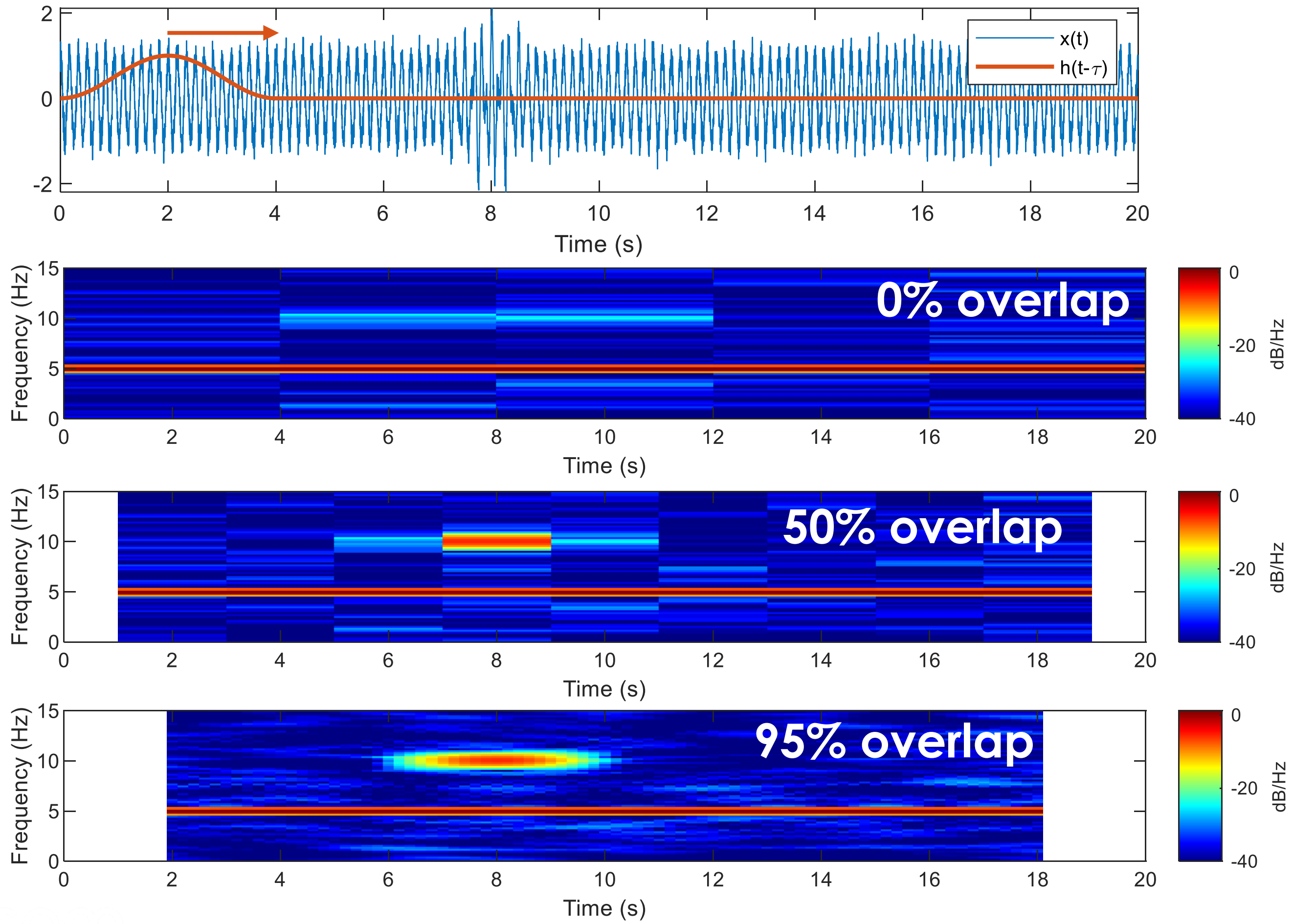
Spectrograms in practice: Main parameters

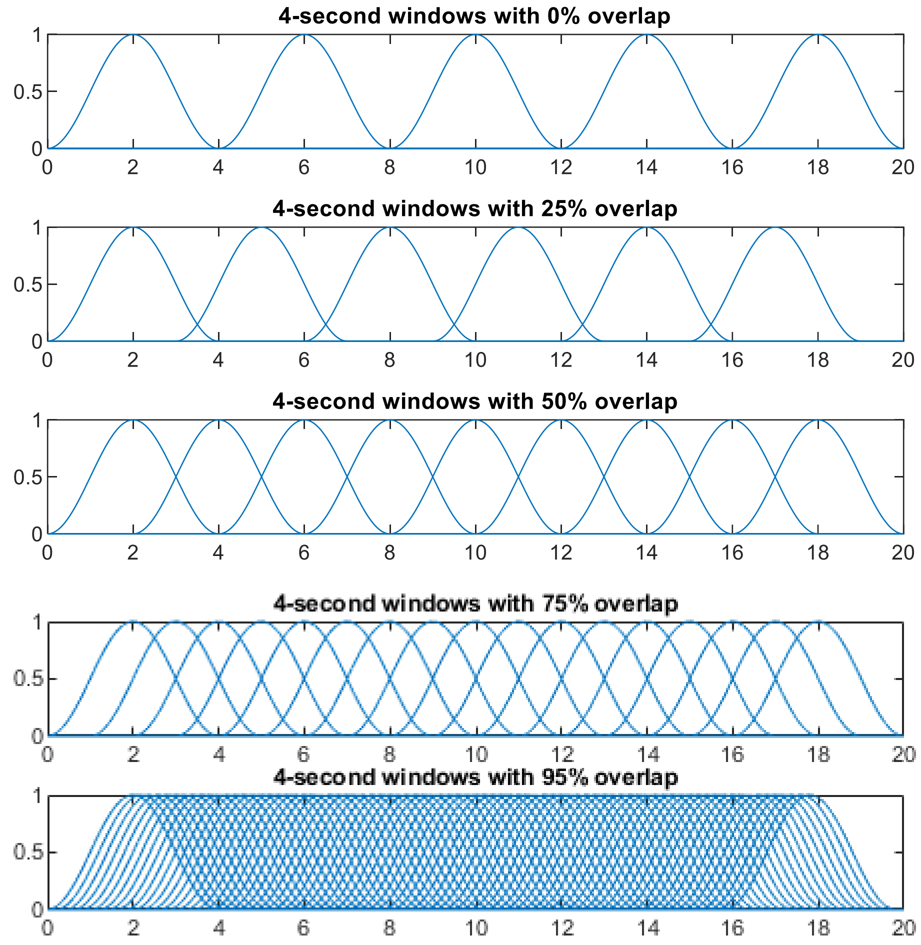
Overlap

- Higher overlap → avoids missing transient phenomena, but requires a higher computational cost

The overlap determines the sampling rate of the time variable of the STFT

$$X_h(\tau, f) = \int_{-\infty}^{\infty} x(t)h(t-\tau)e^{-i2\pi f t}dt$$

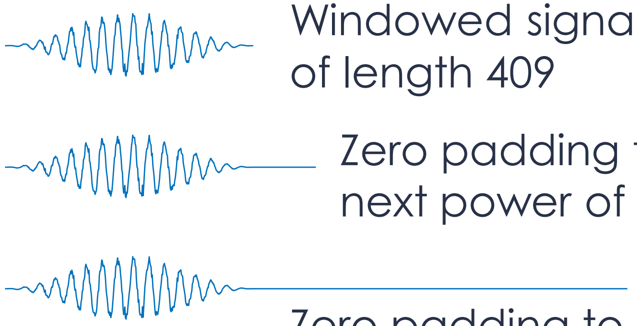




EPFL csem

Spectrograms in practice: Main parameters
Number of FFT points

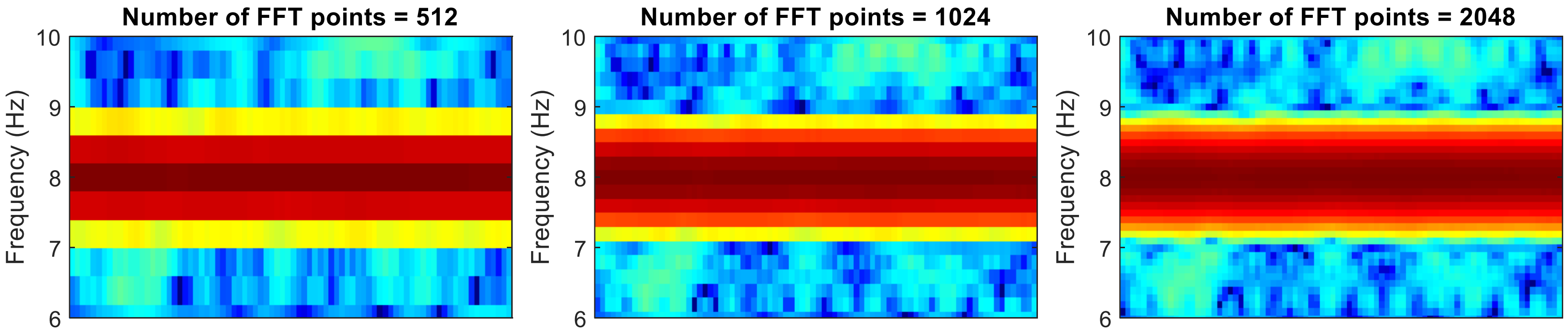
- Discrete Fourier transform (DFT)
  - (Windowed) signal x is N samples long → DFT(x) is N samples long
- Implementation using the FFT
  - x is zero-padded so that N is the next power of 2
  - Increasing the zero padding (using an even larger power of 2) improves virtually the frequency resolution but does not add new information



Windowed signal x of length 409

Zero padding to the next power of 2 (512)

Zero padding to an even larger power of 2 (1024)



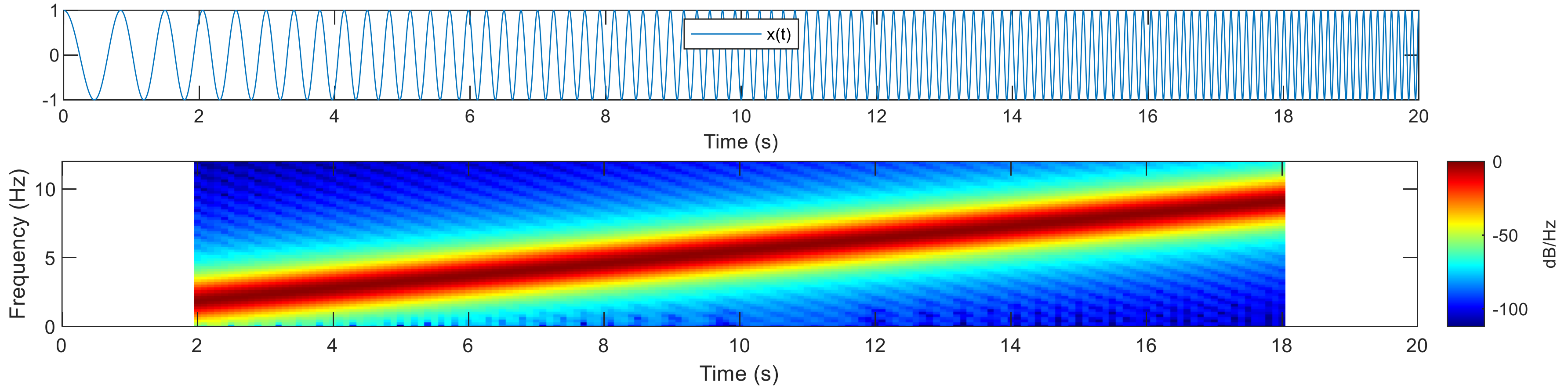
Spectrograms in practice: Main parameters

- In Matlab: `spectrogram(x, window, noverlap, nfft, fs)`
- `window:` window function or window length using default window function
  - `hann(512)` → Uses a Hann window of length 512
  - `512` → Uses a Hamming window (Matlab's default) of length 512
  - `[]` → Uses a Hamming window with a length such that `x` is divided into 8 segments
- `noverlap:` number of overlapping samples between consecutive windows
  - `492` → Corresponds to 95% overlap for a window of length 512
  - `[]` → Uses a number that produces 50% overlap (Matlab's default)
- `nfft`: number of FFT points or vector of a specific frequency range
  - `2^11` → Zero pads each window such that it uses 2^{11} points for the FFT
  - `[]` → Uses the next power of 2 equal or higher than the window length, but at least 256 (2^8) points
  - `linspace(2, 5, 2^11)` → Specific frequency range. Matlab's default: `linspace(0, fs/2, nfft/2+1)`

Spectrograms in practice: Main parameters

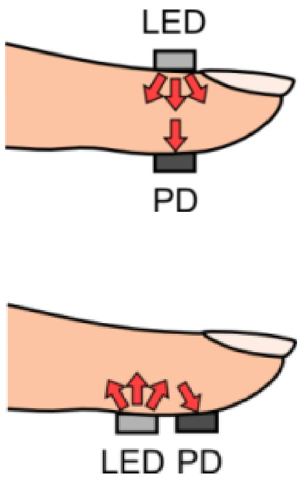
Usage example: Suppose we have a 20-second-long chirp signal sampled at 400 Hz and we want to use windows of 4 seconds with 97.5% overlap to analyze it

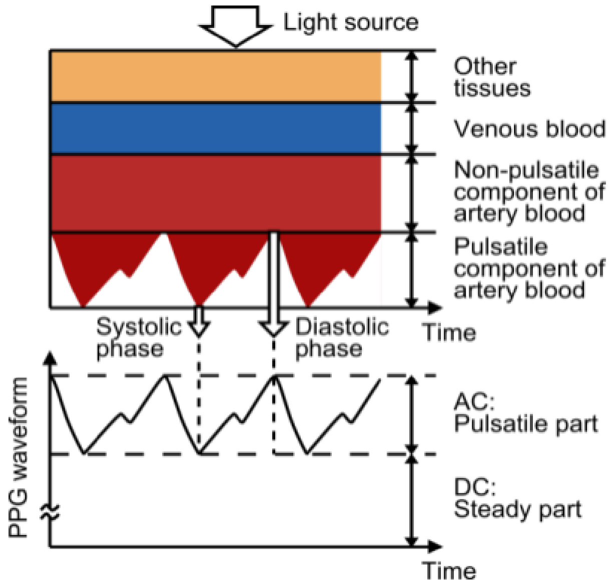
```
fs = 400;                               % Sampling frequency in Hz
t = (0:1/fs:20-1/fs)';                  % 20-second time vector
x = chirp(t, 1, t(end), 10);            % Linear chirp from to 1 to 10 Hz
window = round(4 * fs);                 % Window length (1600 samples)
noverlap = round(0.975 * 4 * fs);       % Overlap between consec. windows (1560 samples)
nfft = [];                              % Default number of FFT points (2^11 = 2048)
spectrogram(x, window, noverlap, nfft, fs, 'yaxis'); % 'yaxis': frequencies on y axis
```

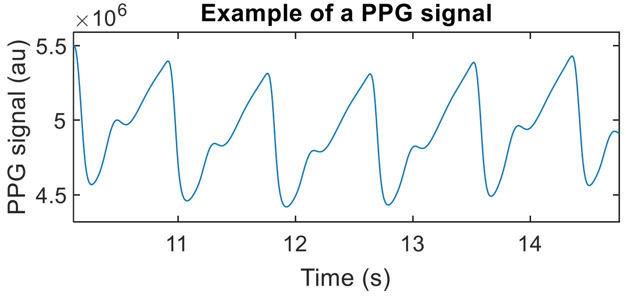
Example: PPG signal in anesthesia

- Photoplethysmography (PPG)
  - An LED emits light through the finger and a photodiode (PD) measures the portion of the light that has not been absorbed by the tissue. At each heartbeat, the pressure wave generated by the heart dilates the small vessels in the finger. More light is absorbed, thereby creating pulses in the PPG signal measured by the PD.
  - Routinely used in clinical practice for the non-invasive monitoring of oxygen saturation and heart rate.

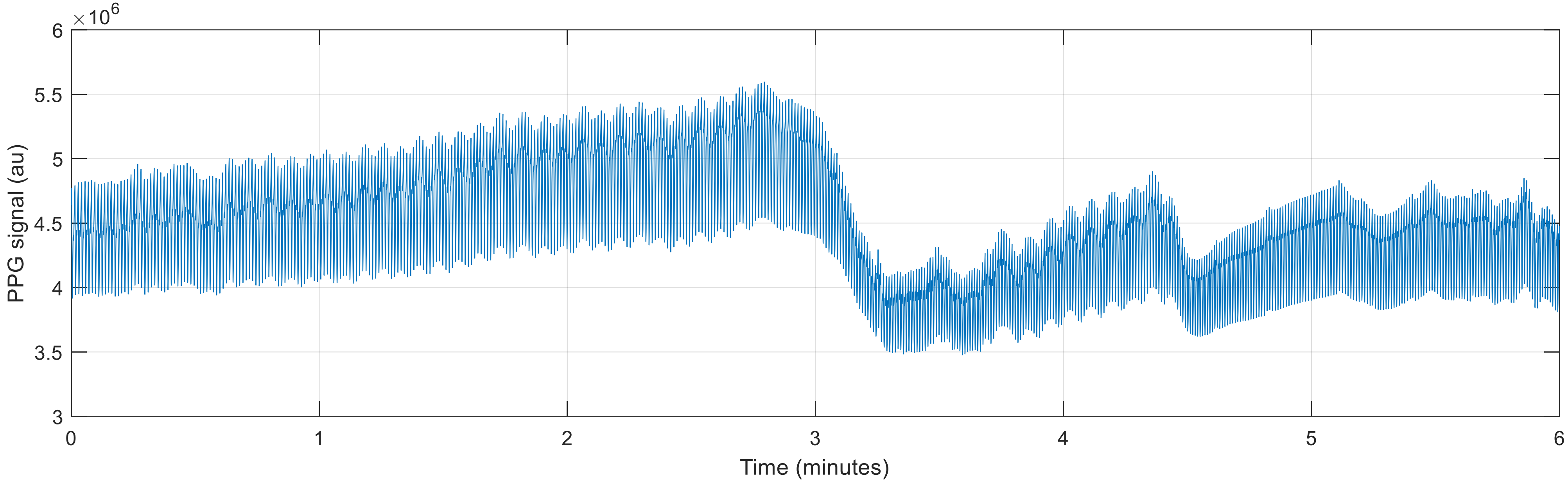


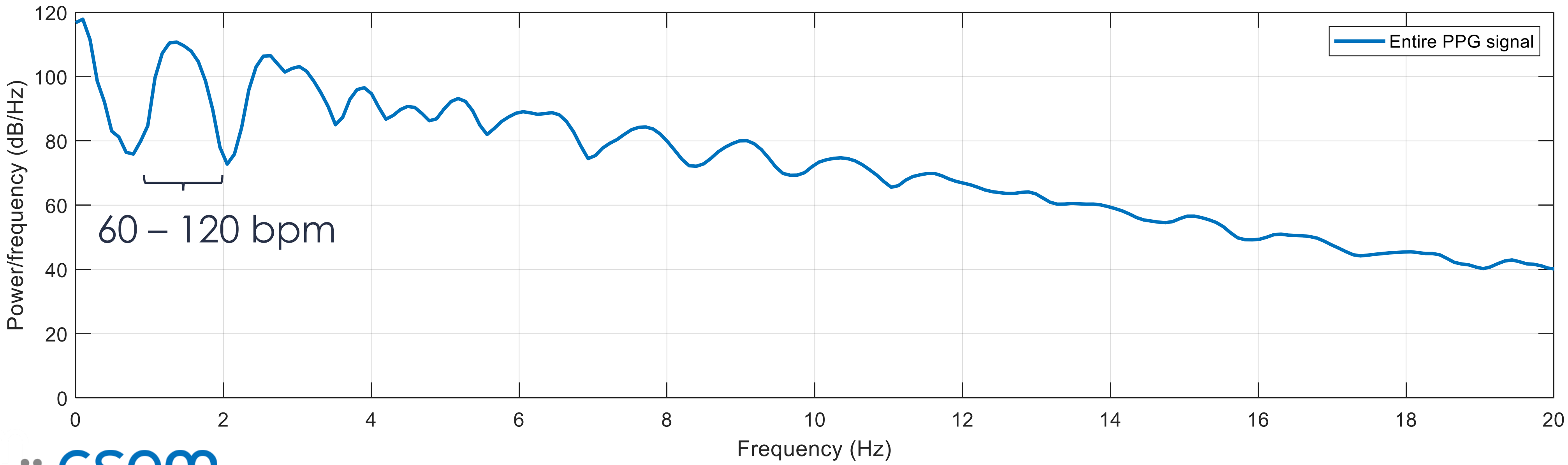


Tamura T, et al. *Electronics*. 2014; 3(2):282-302

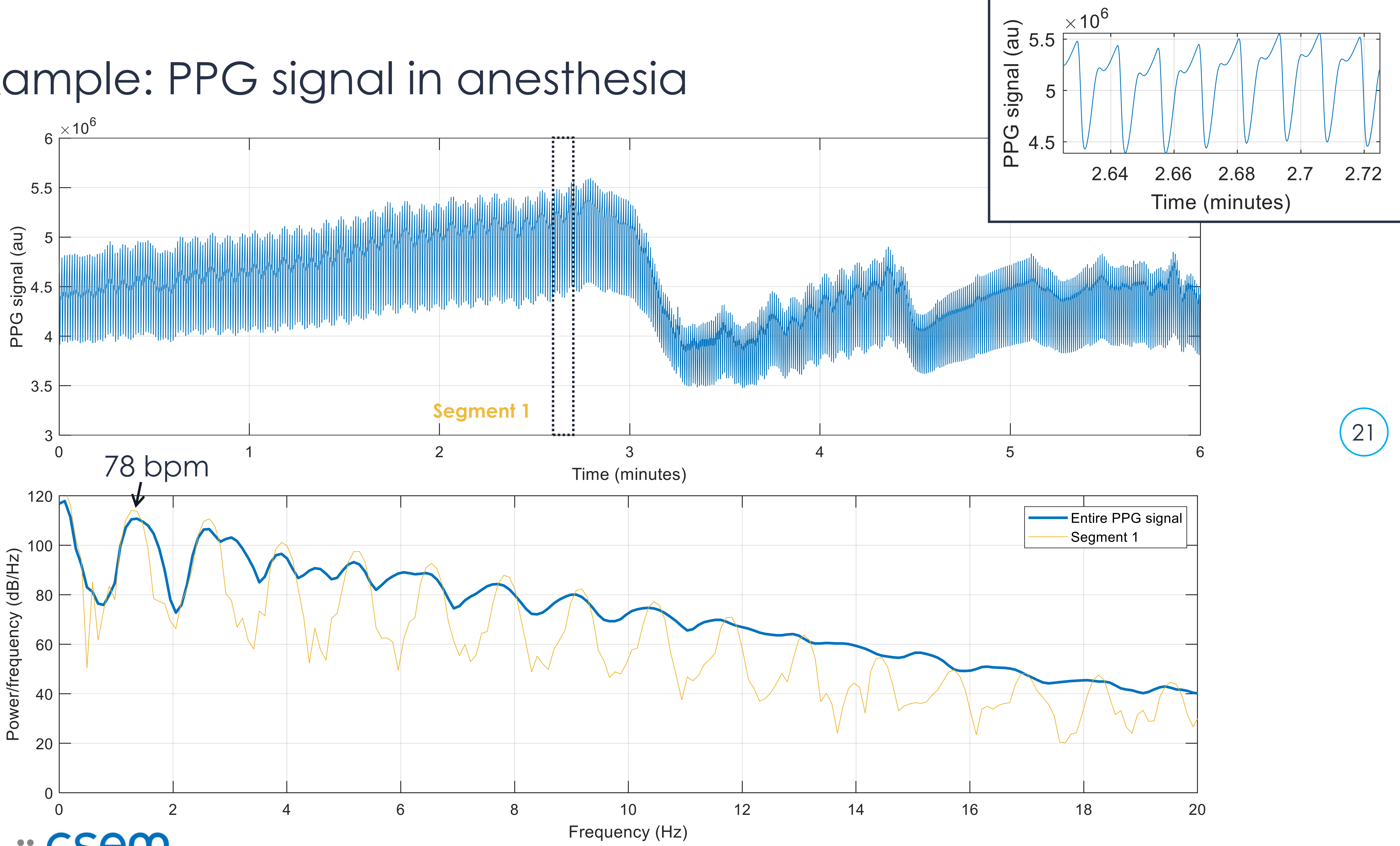


Example: PPG signal in anesthesia



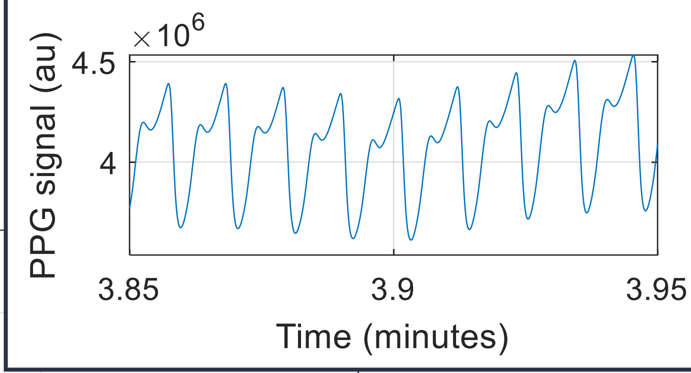


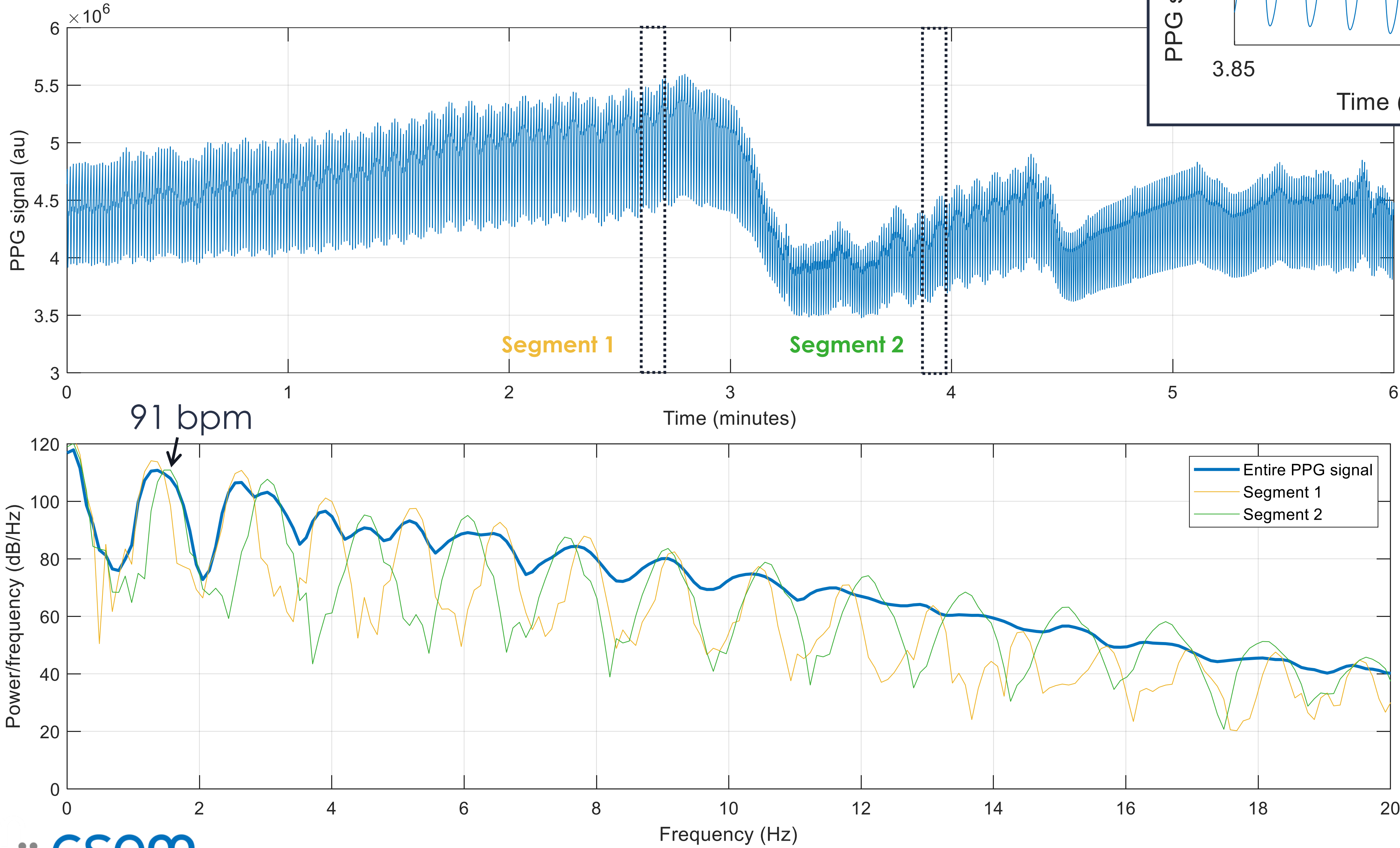
Example: PPG signal in anesthesia



EPFL csem

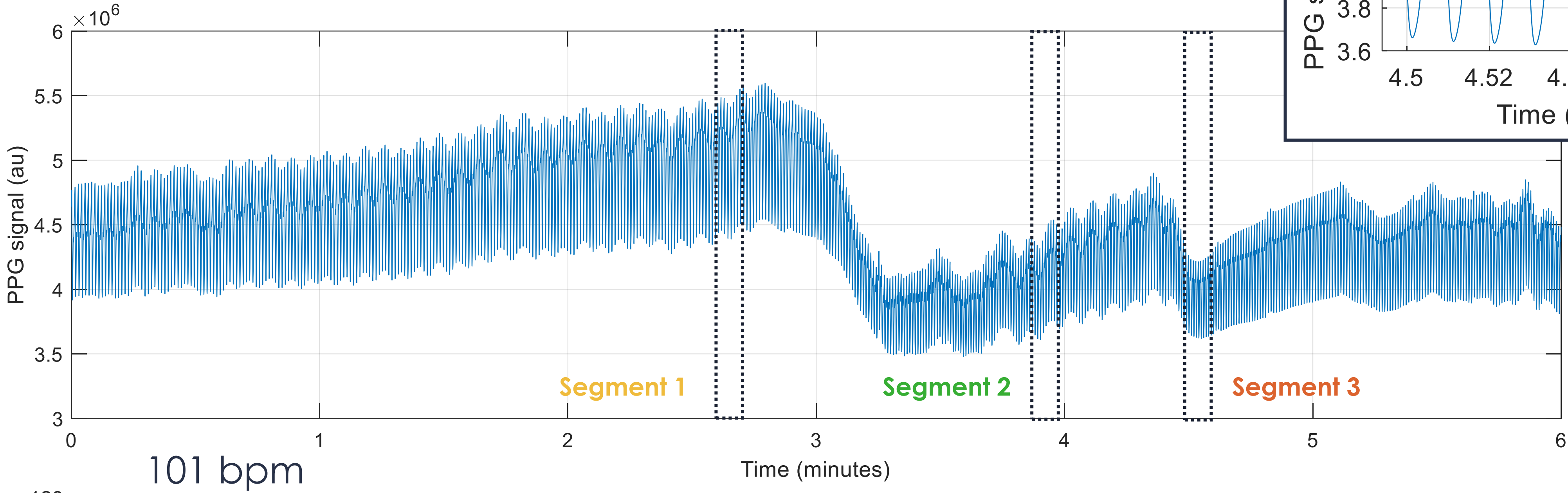
Example: PPG signal in anesthesia

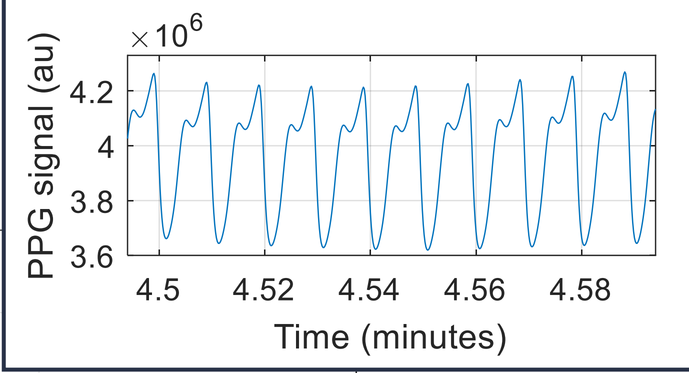


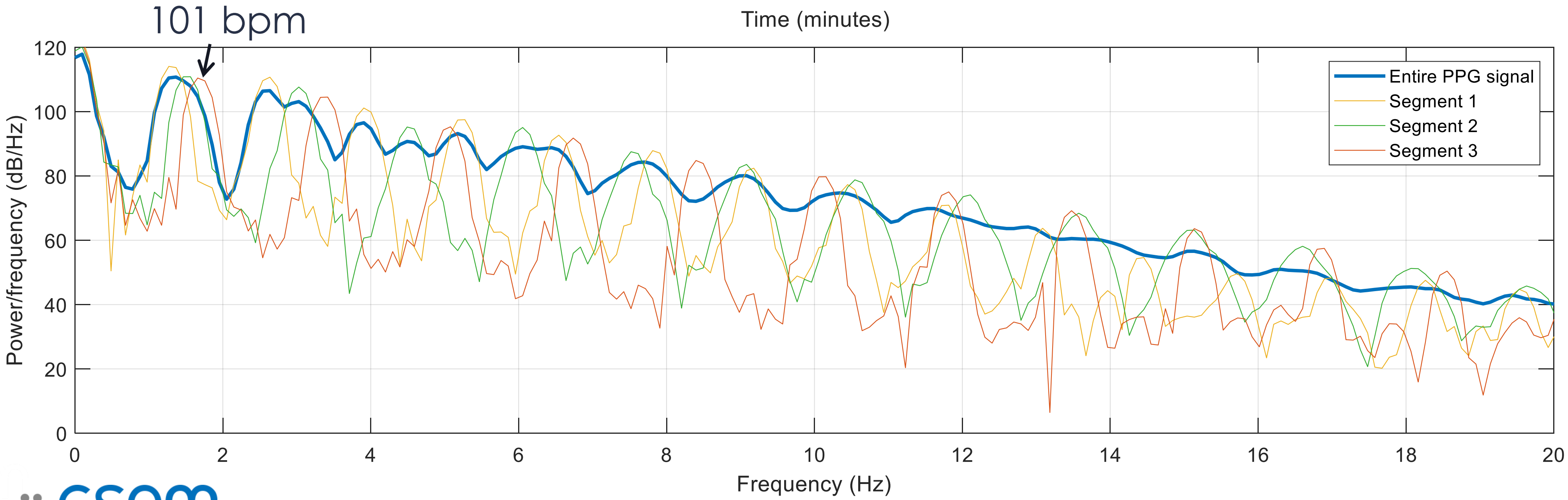


EPFL csem

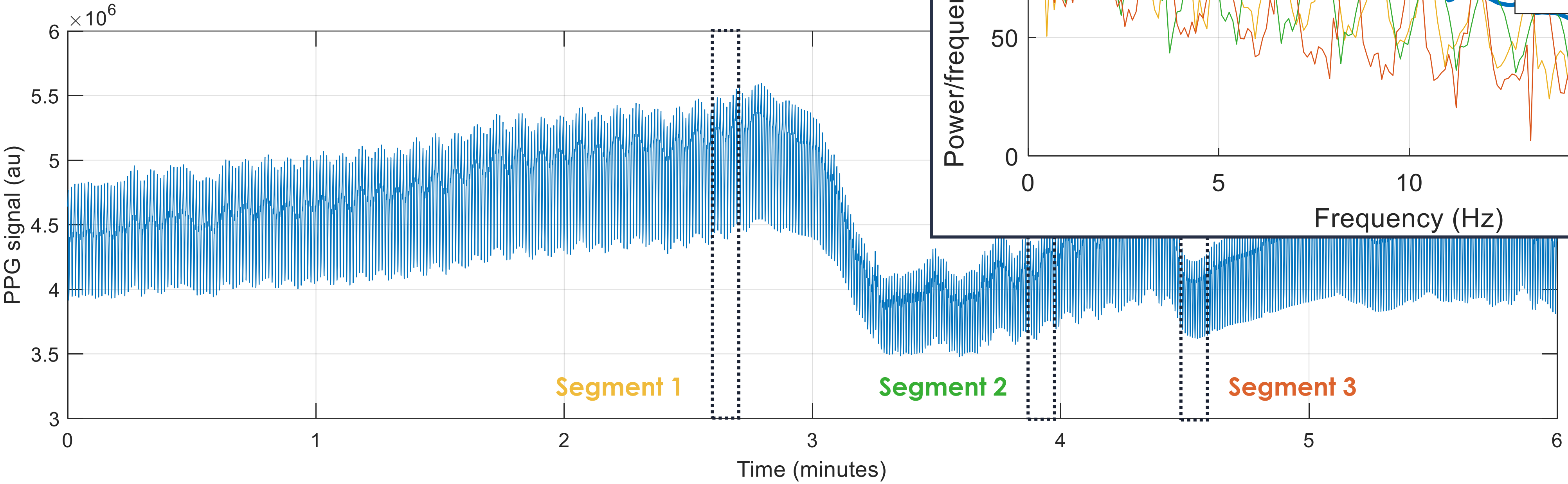
Example: PPG signal in anesthesia

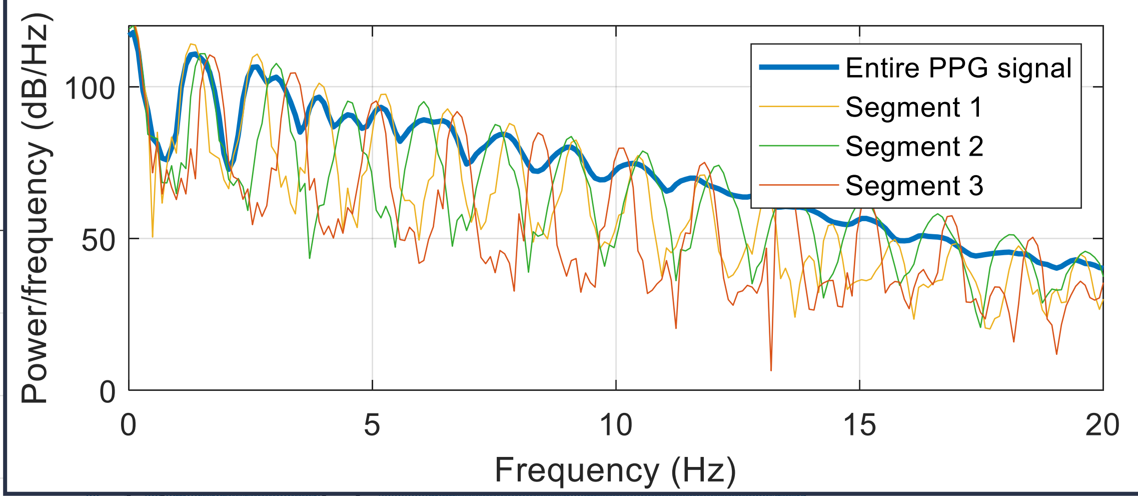


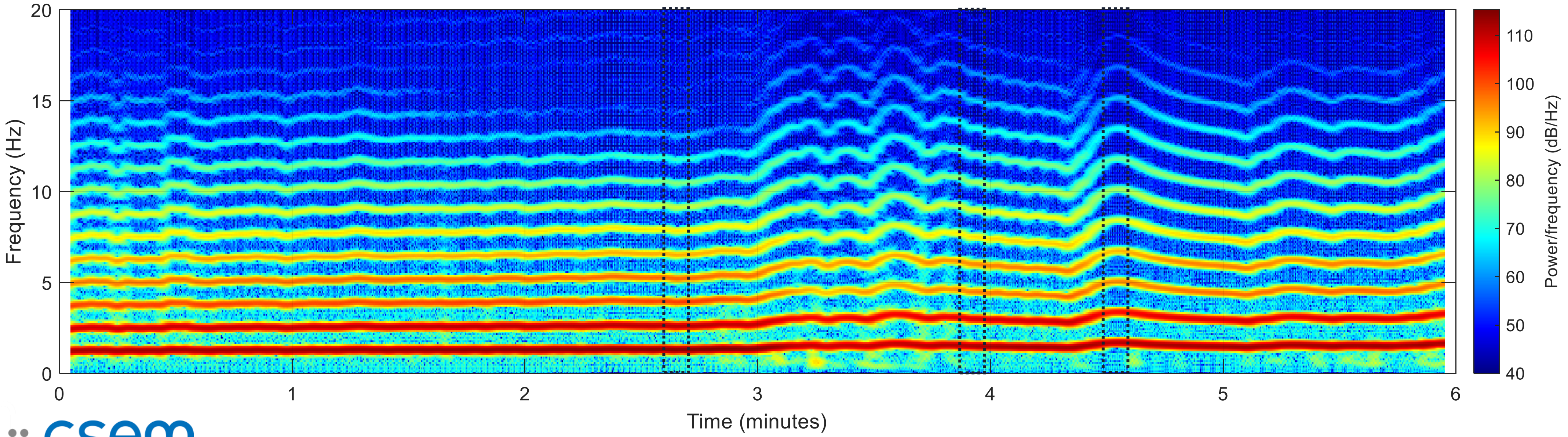




Example: PPG signal in anesthesia



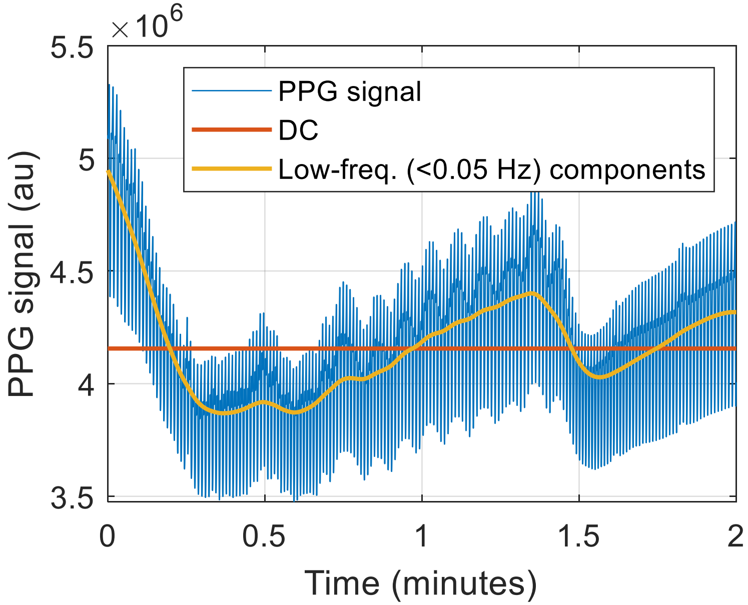




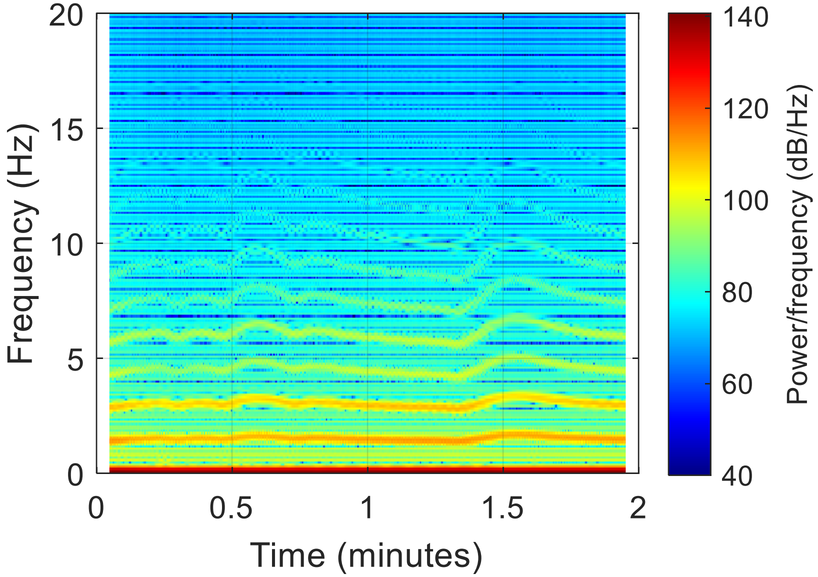
EPFL csem

Spectrograms in practice: Baseline attenuation

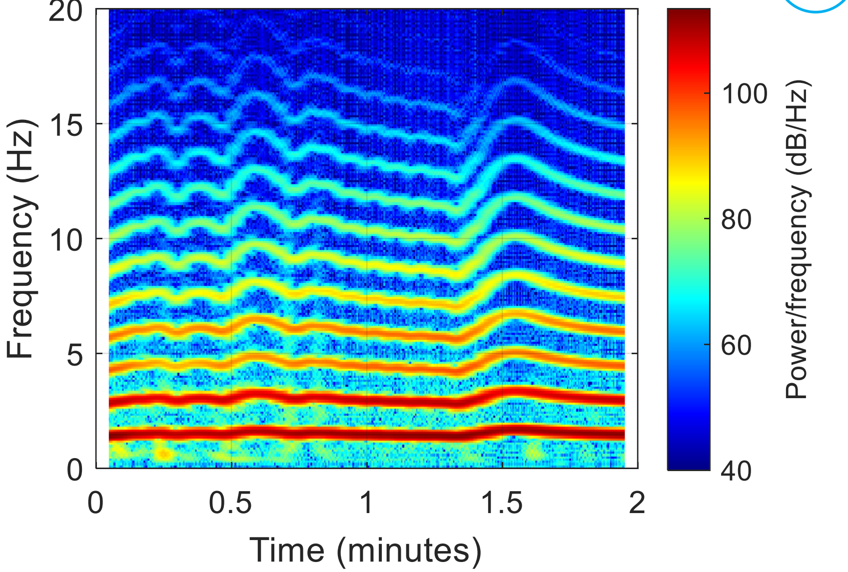
- In many biomedical applications, the DC (0 Hz) component – or the low frequency components in general – are of little to no interest, but are unfortunately of large magnitude, and therefore ‘drown’ lower-amplitude frequency components in the spectral leakage of the baseline



No baseline attenuation



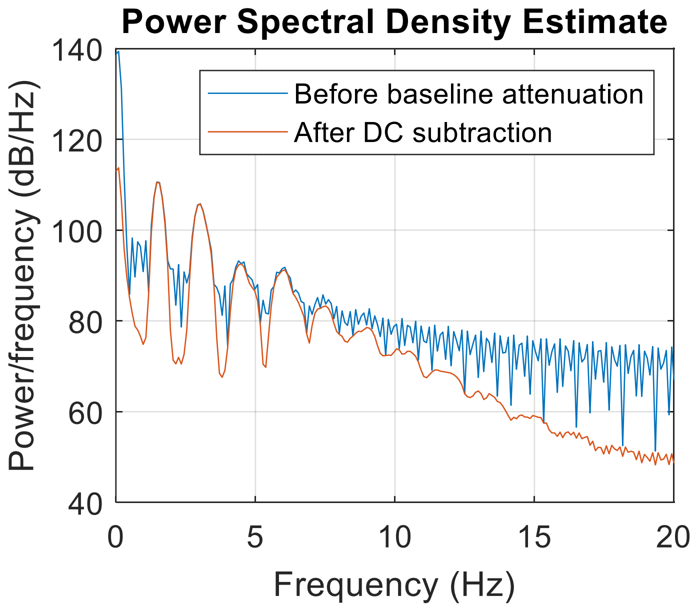
With baseline attenuation

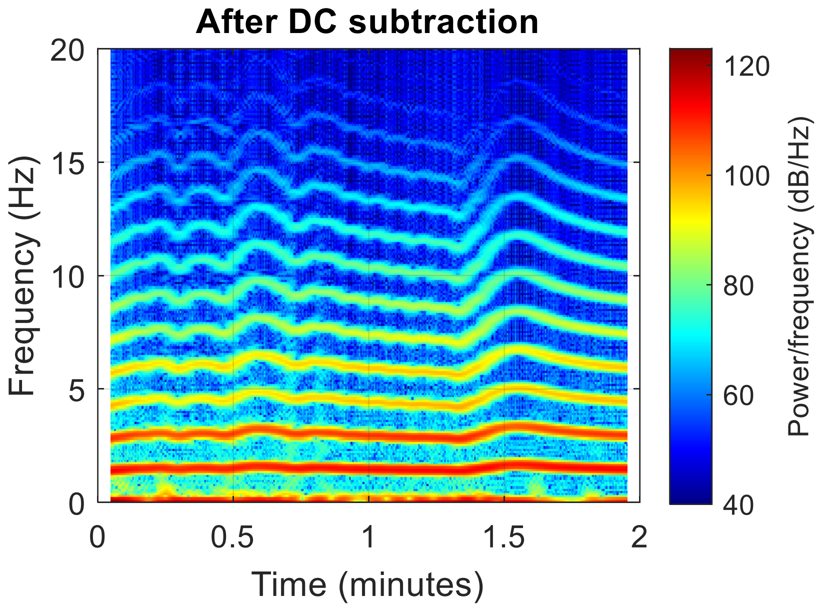


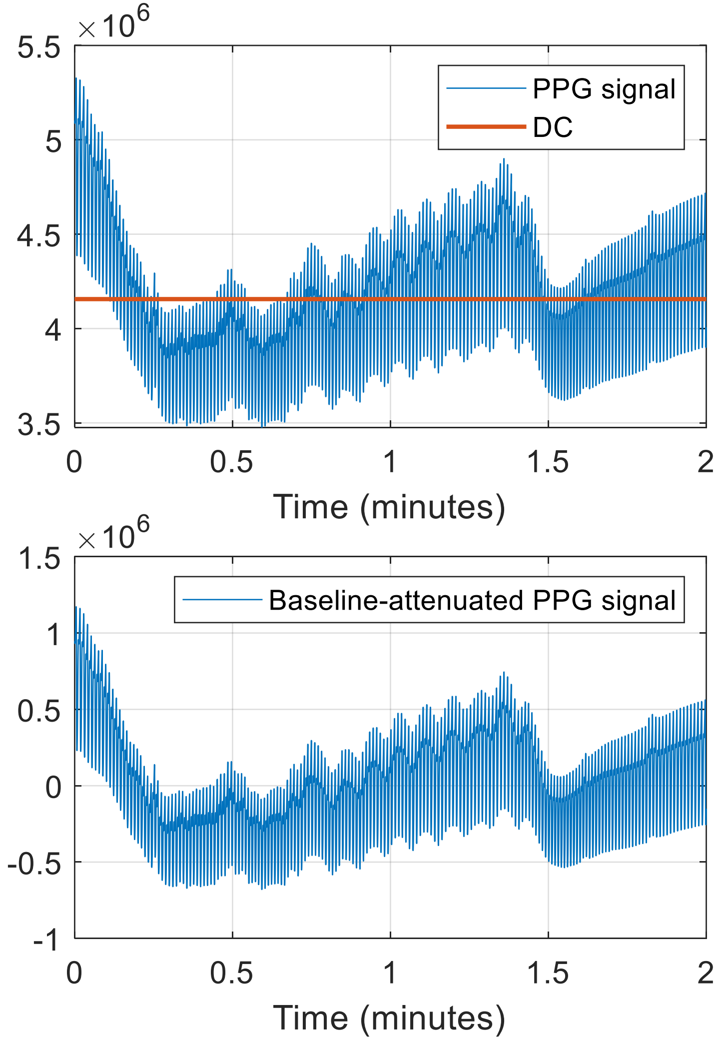
Spectrograms in practice: Baseline attenuation

- **DC subtraction:** Simplest, but not always sufficient

```
x = x-mean(x);
```





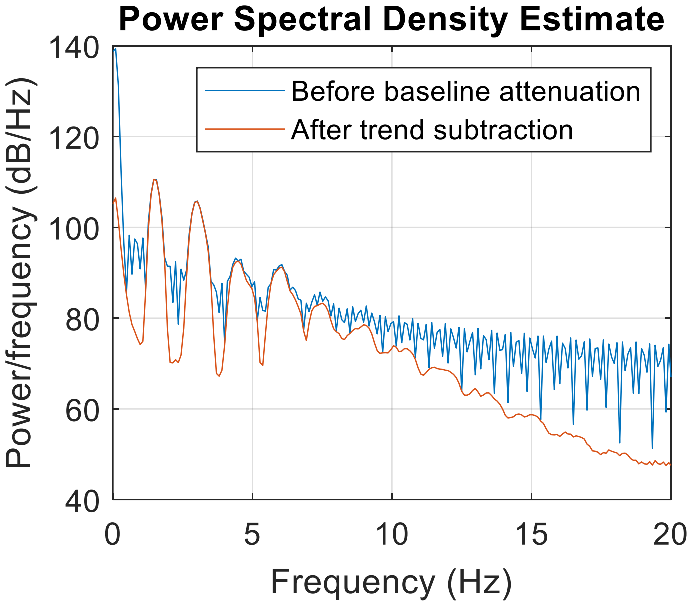


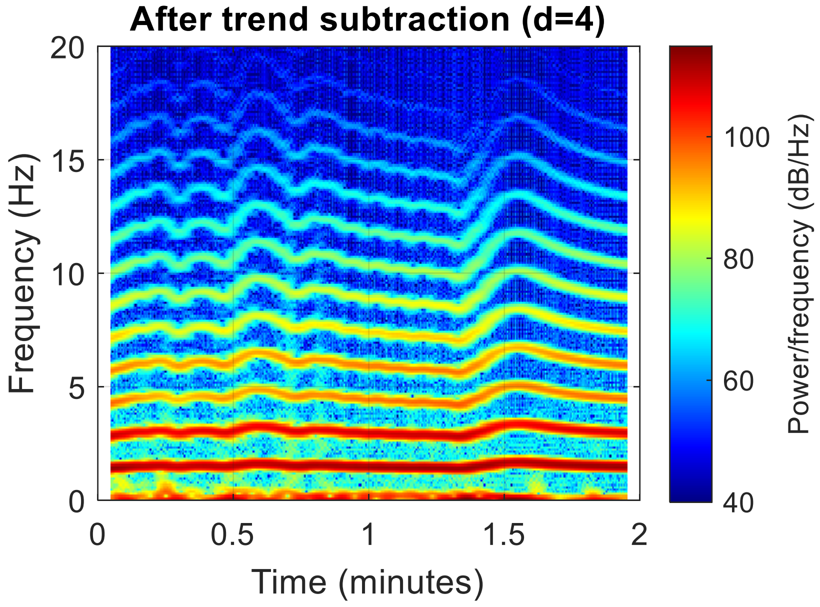
EPFL csem

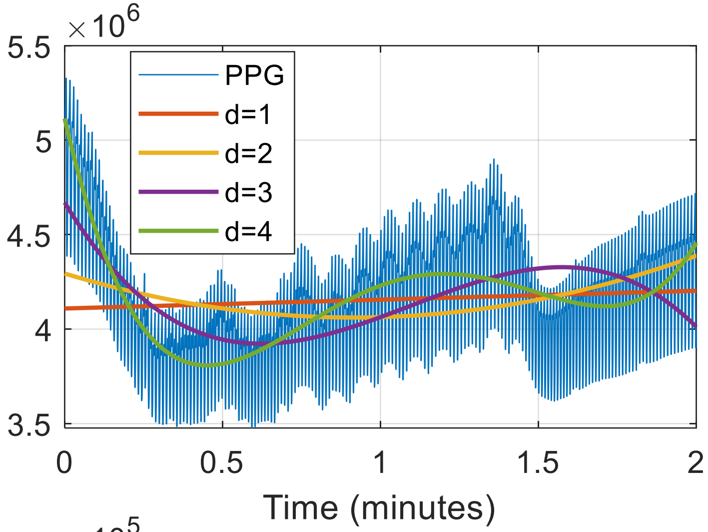
Spectrograms in practice: Baseline attenuation

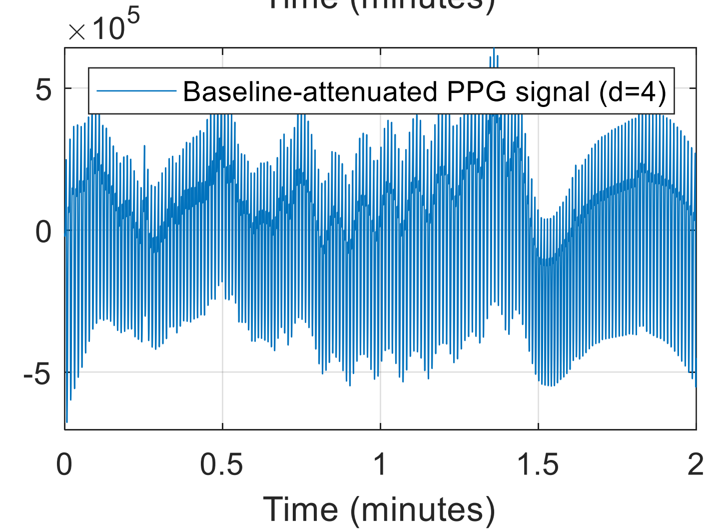
- **Trend subtraction:** Generally sufficient, but requires careful supervision (visual inspection) for higher-order fits that may lead to unexpected results

```
d = 4;  % Degree of the polynomial fit
x = detrend(x,d);
```







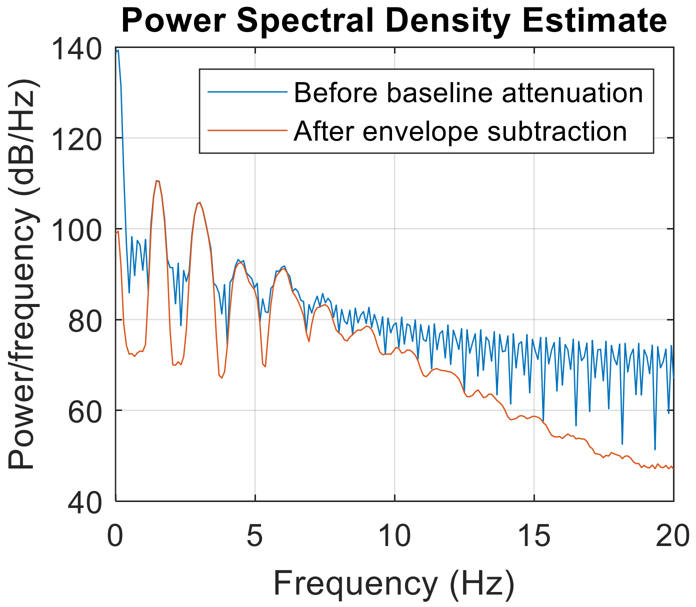


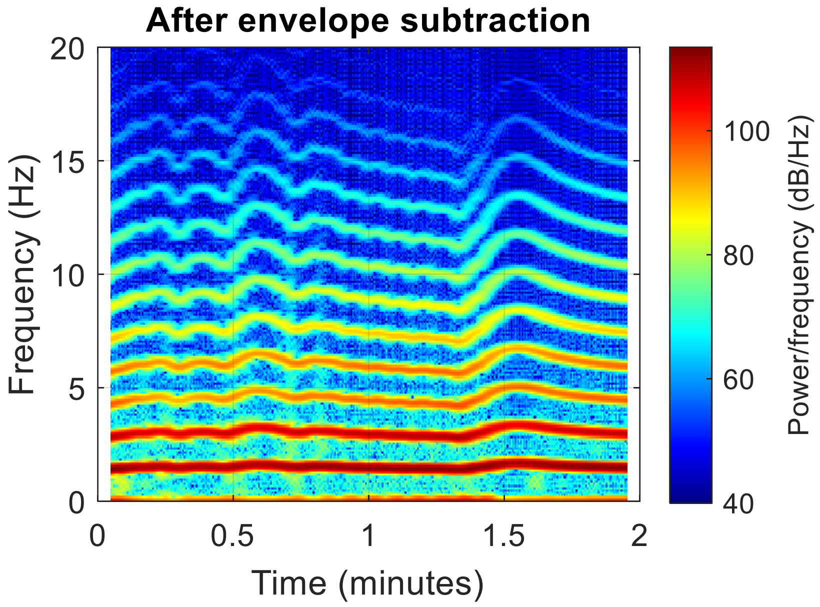
EPFL csem

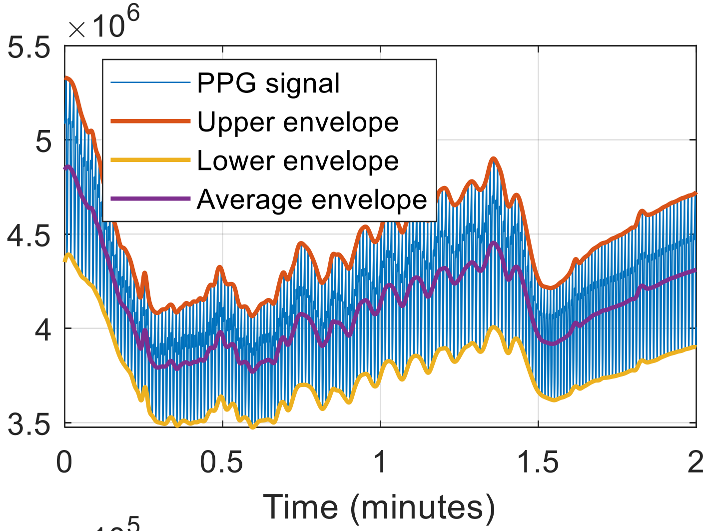
Spectrograms in practice: Baseline attenuation

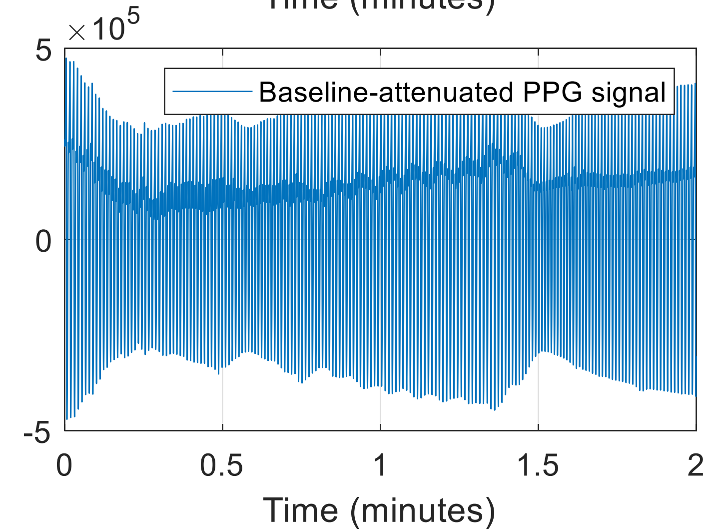
- **Average envelope subtraction:** Efficient, but requires careful supervision (visual inspection) for a proper parameterization and to avoid unexpected results

```
[xu,xl] = envelope(x, round(0.4*fs), 'peak');
x = x-(xu+xl)/2;
```





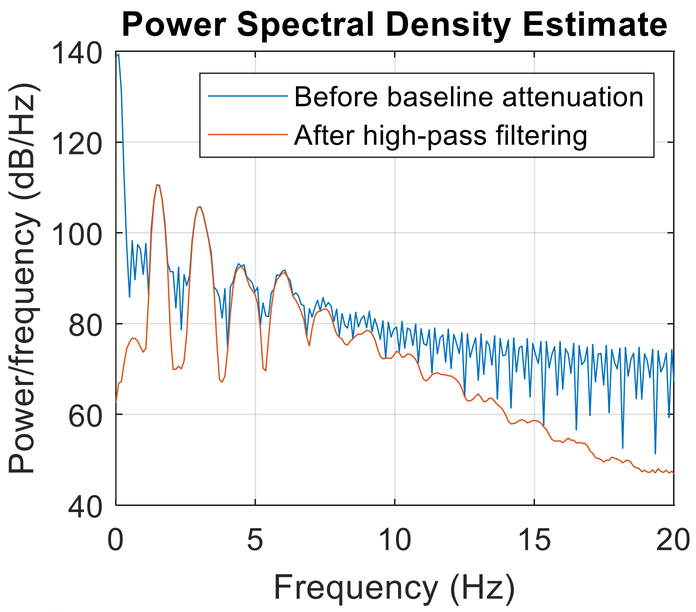


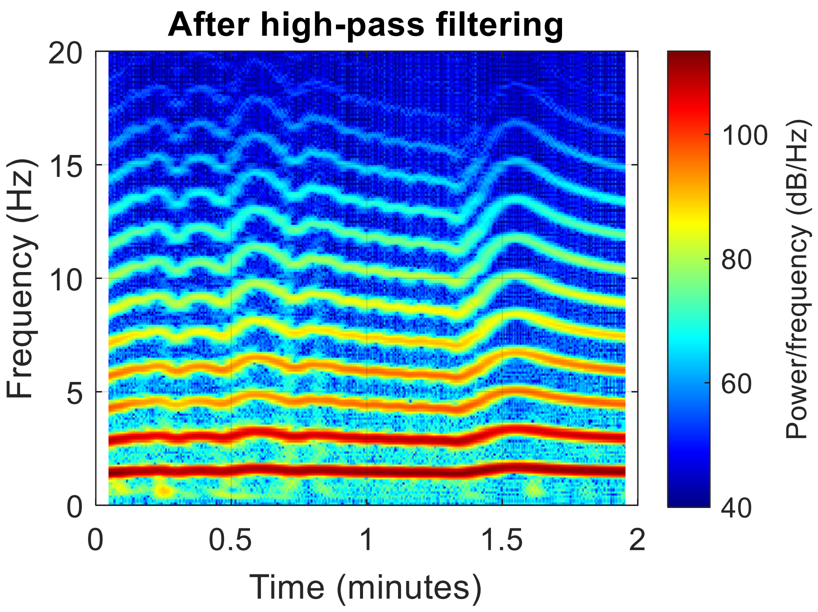


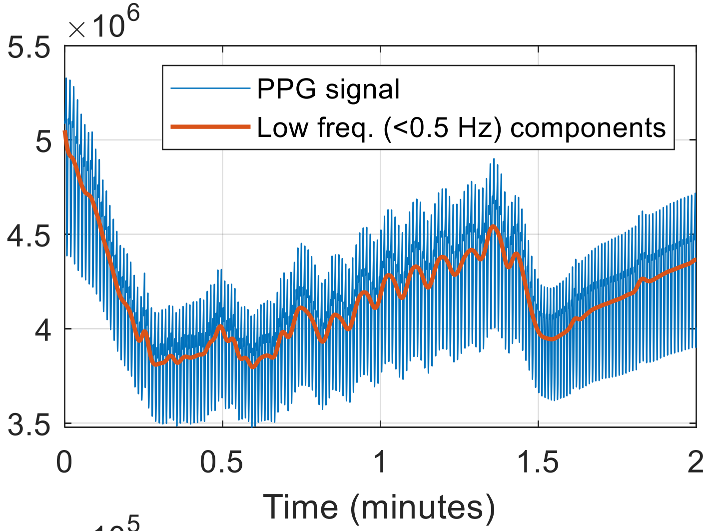
Spectrograms in practice: Baseline attenuation

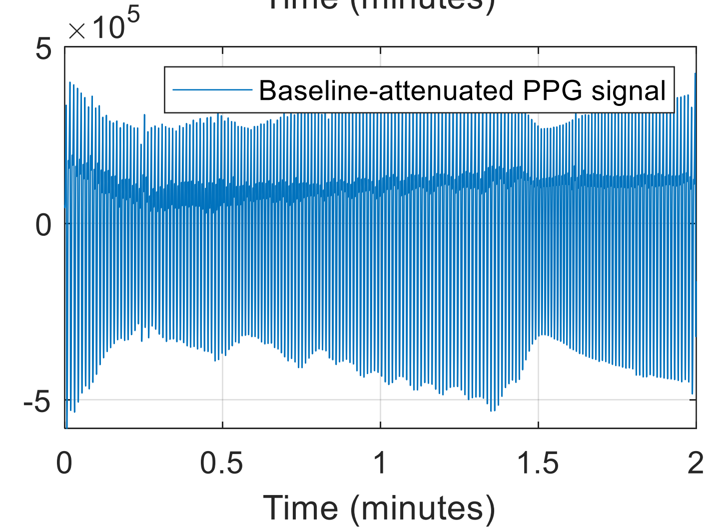
- **Filtering:** Efficient, but requires prior knowledge about the signal to set the cut-off frequency. For instance, for PPG-based heart rate monitoring, one can consider a minimal physiological heart rate of 30 bpm (0.5 Hz).

```
[b,a] = butter(2, 0.5/(fs/2), 'high');
x = filtfilt(b,a,x);
```



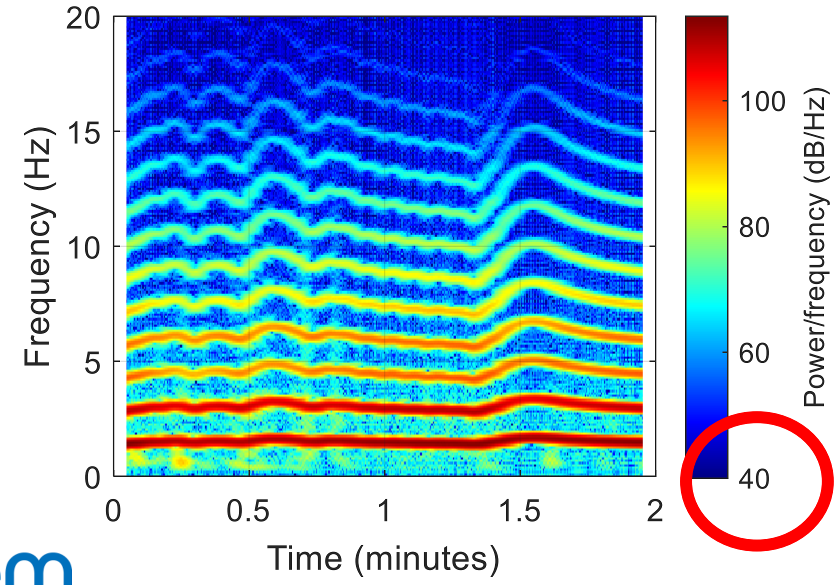


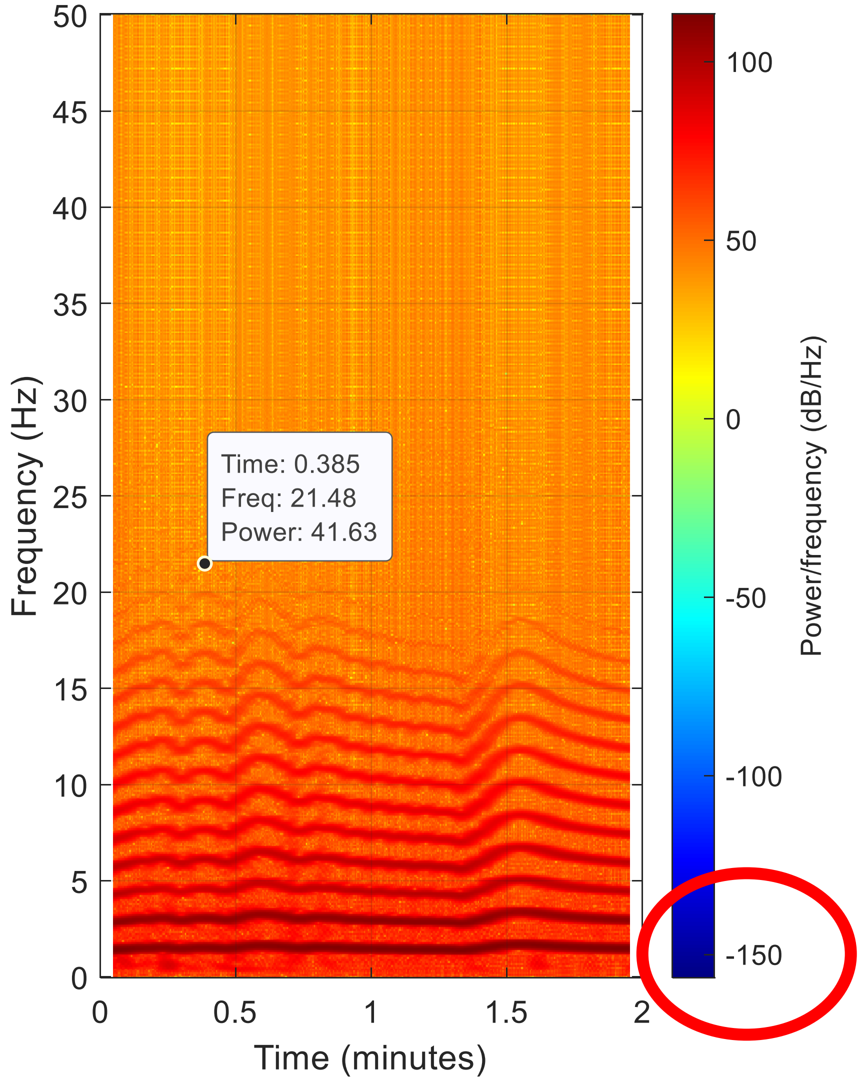




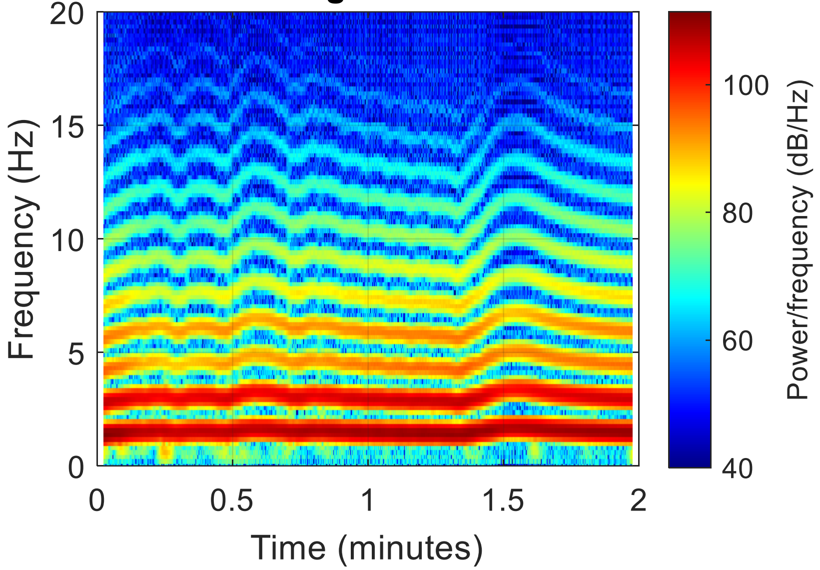
Spectrograms in practice: Limitation of the colormap range

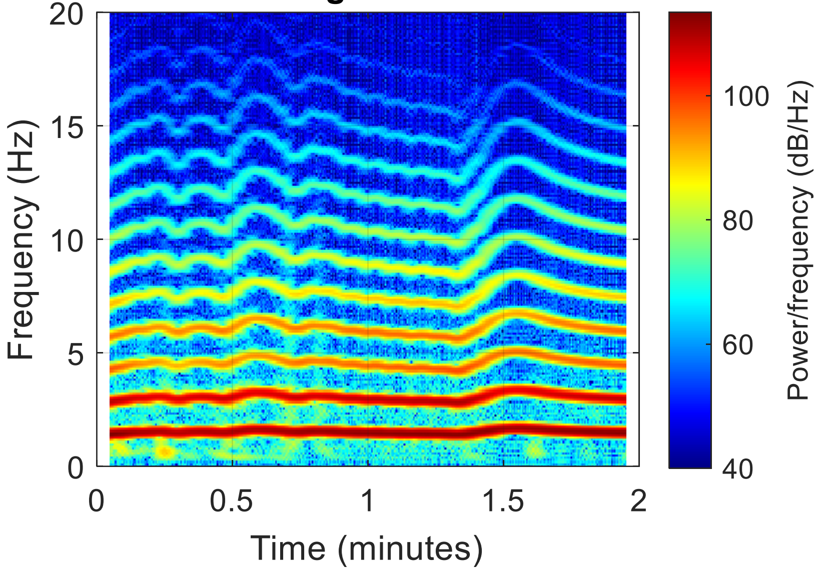
- Without limitation of the colormap range, the spectrogram of the baseline-attenuated PPG signal actually looks like this ⟹
- The highest (barely visible) harmonic of the PPG signal has a power of 41.63 dB/Hz → 40 dB/Hz is a good threshold choice to consider anything below that value as noise

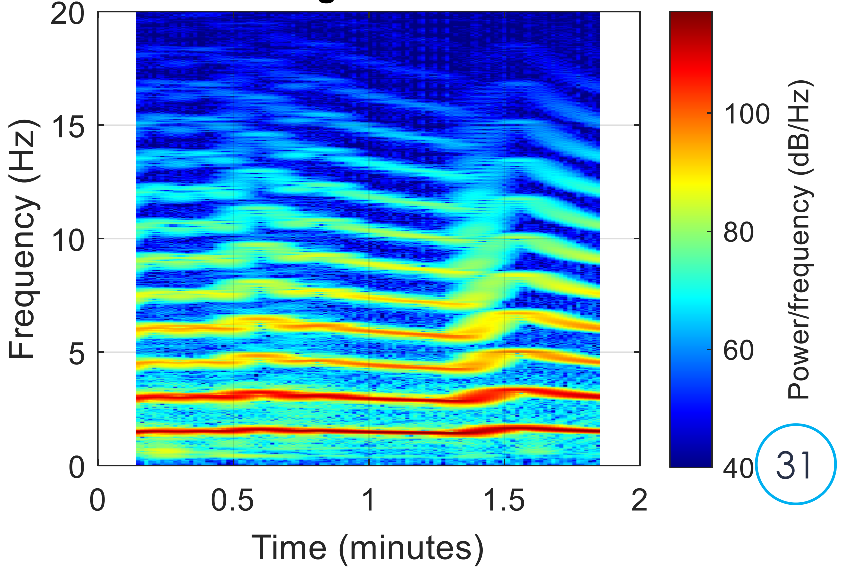


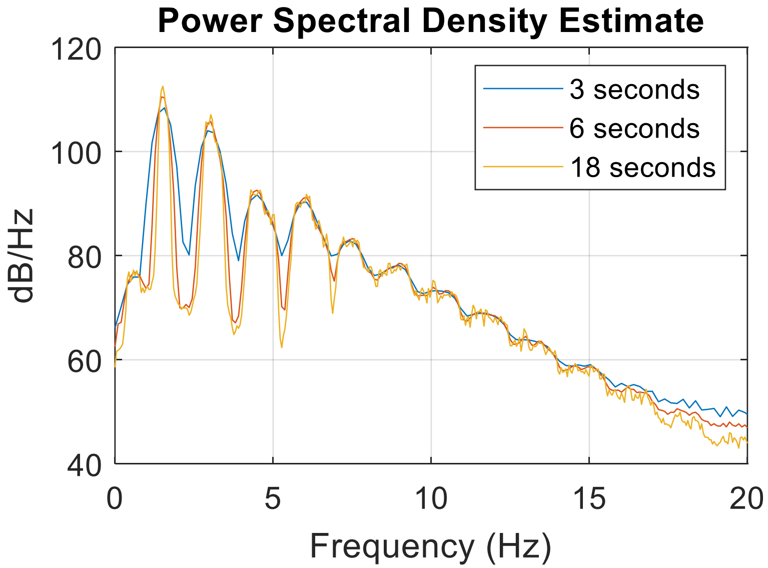


Spectrograms in practice: Window length influence



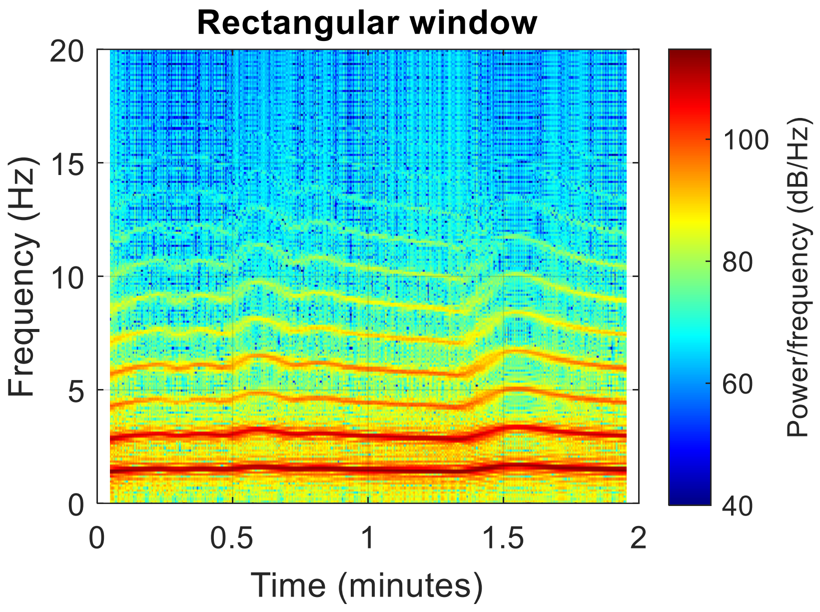


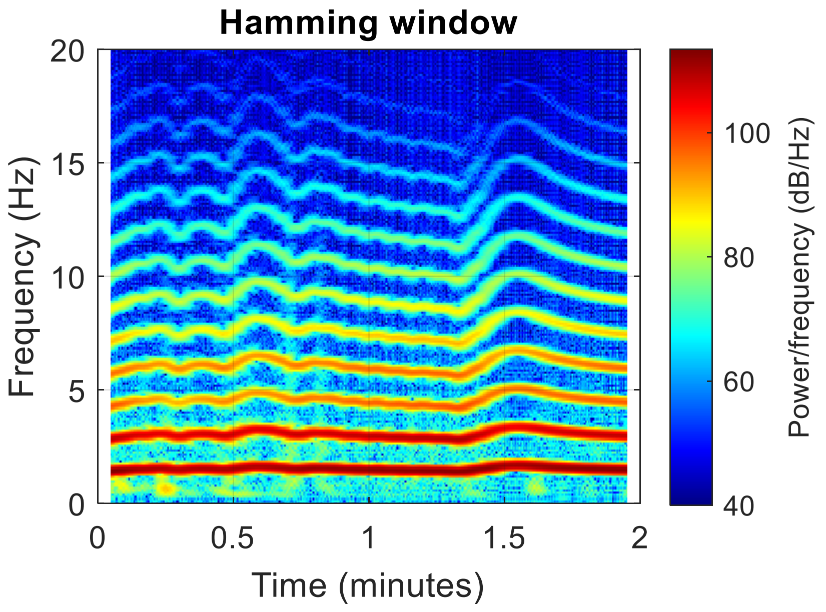


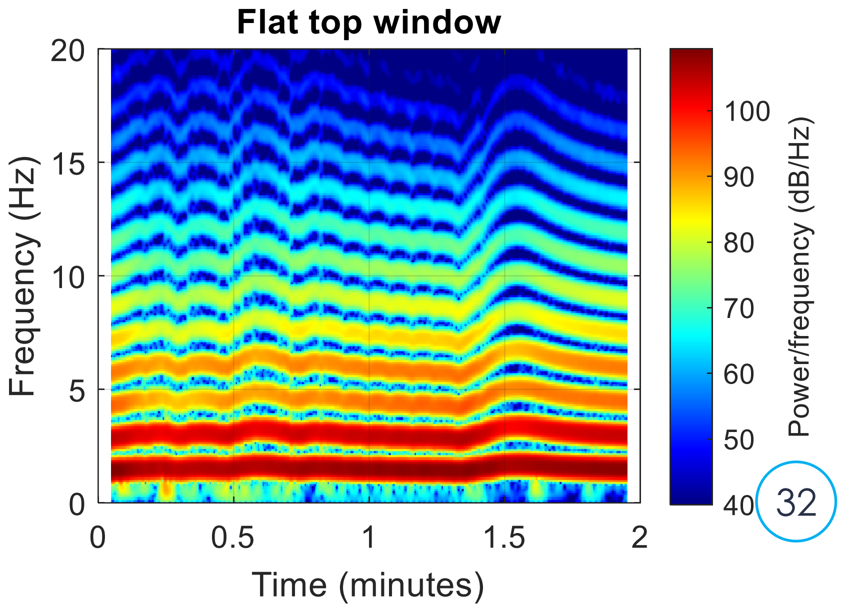


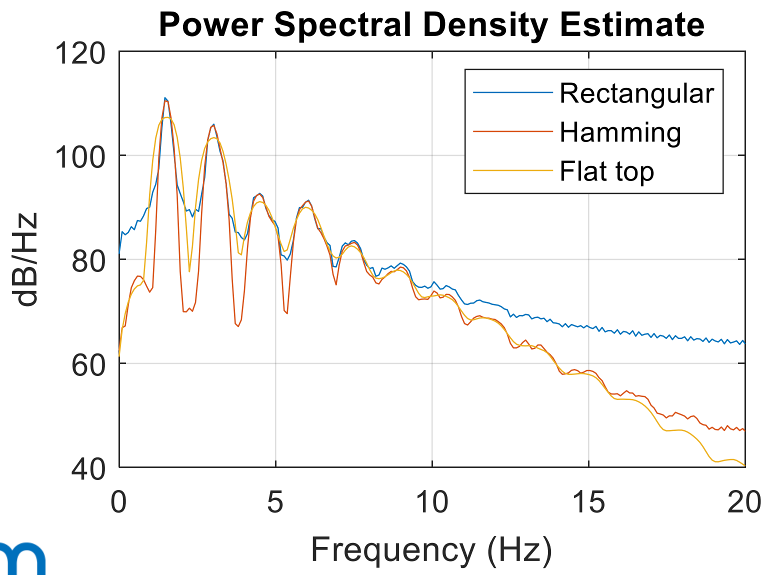
EPFL csem

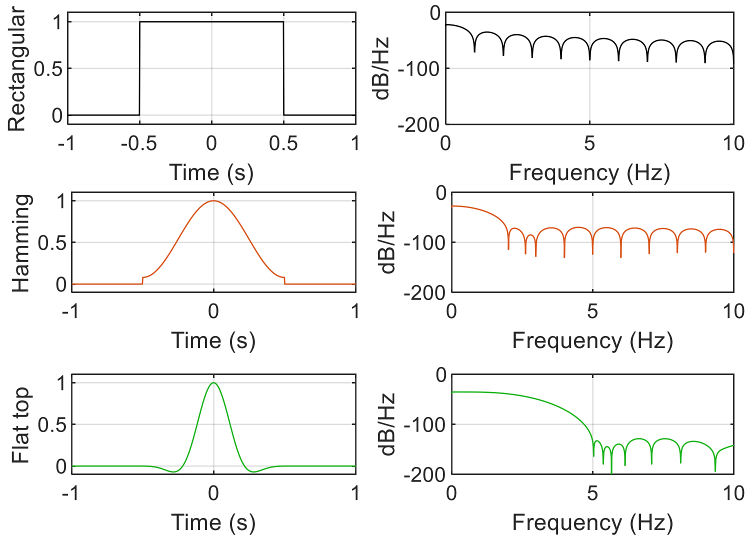
Spectrograms in practice: Window function influence





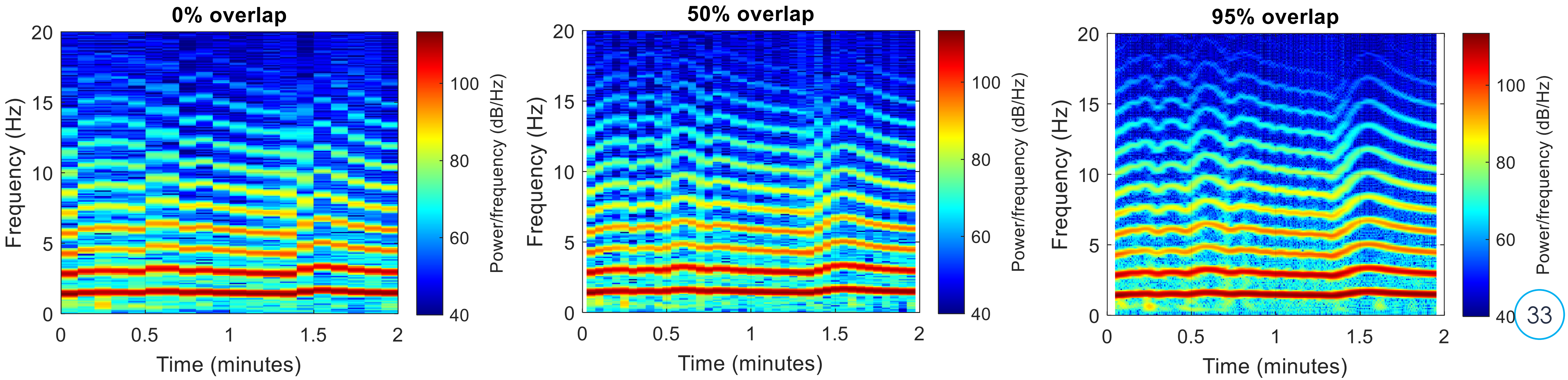


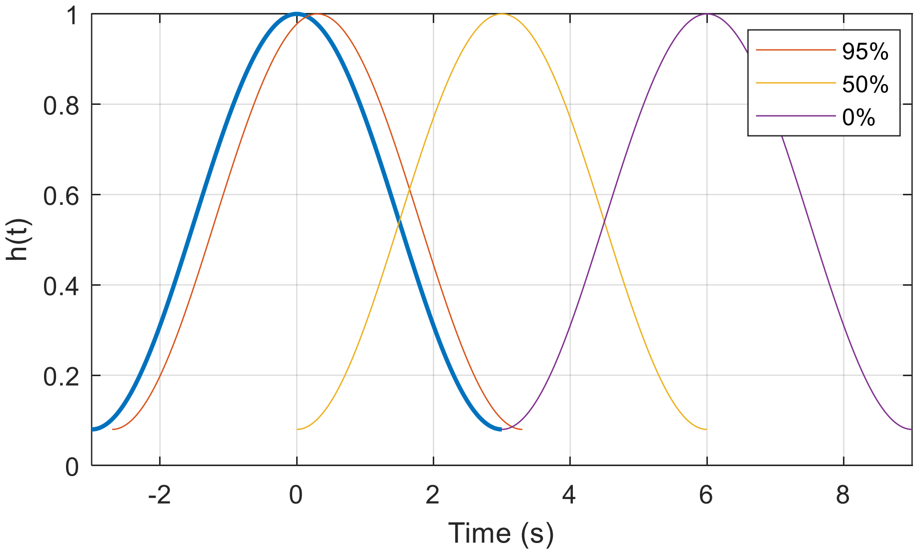




EPFL csem

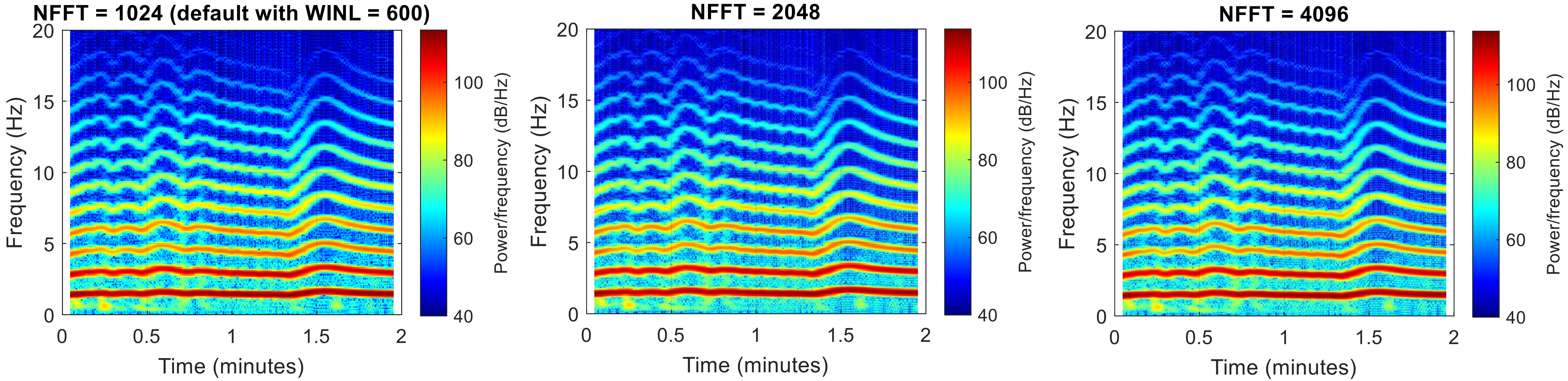
Spectrograms in practice: Overlap influence





EPFL csem

Spectrograms in practice: Number of FFT points

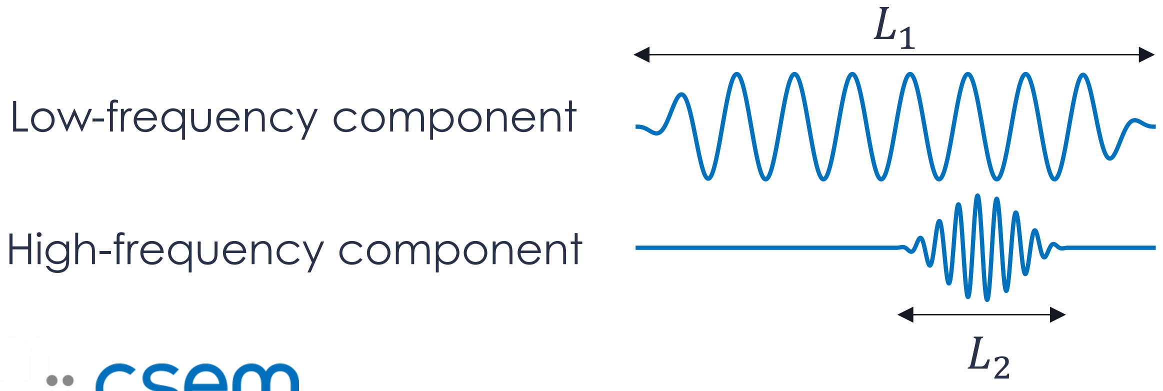


Spectrograms in practice: Summary

- ***Baseline attenuation:*** Always **attenuate the low frequencies** of the signal prior to computing the spectrogram, **if they are of no interest**
- ***Colormap range:*** Set the **lower limit** at the **noise level**
- ***Window length:* Trade-off** between good **frequency resolution** (long window) and good **time resolution** (short window)
- ***Window function:* Trade-off** between good **frequency resolution** (e.g. rectangular window) and good **dynamic range** (e.g. flat top window)
- ***Overlap:*** The **higher the better** if computational cost acceptable
- ***Number of FFT points:*** The **higher the better** if computational cost acceptable, but **default value** (next power of 2 ≥ window length) is **usually perfectly sufficient**

Wavelet analysis: Motivation

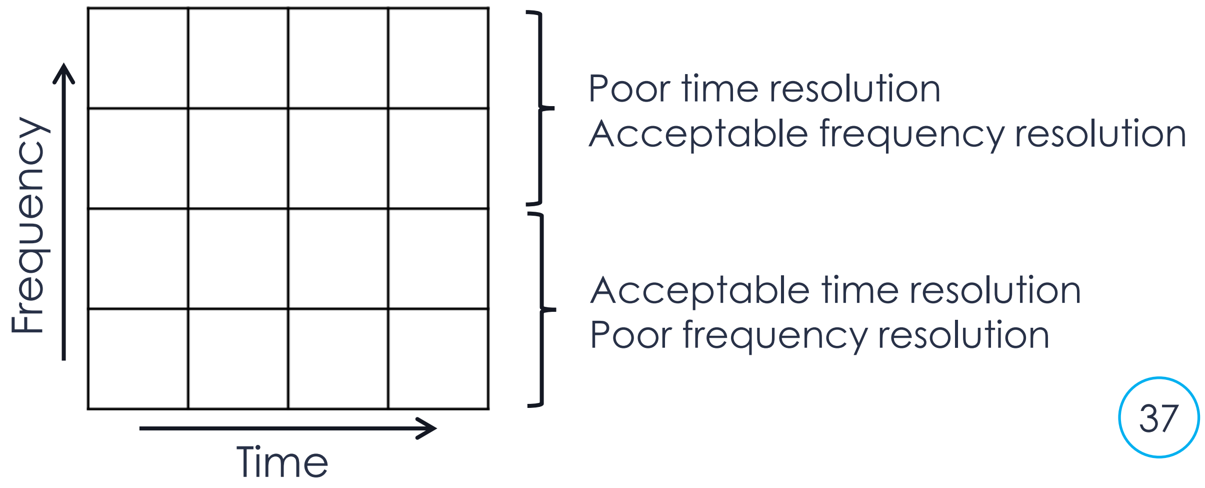
- In real signals, **low frequency** components often span over a longer time than high frequency components: a **longer time window** is therefore **required** to properly capture their frequency
- **High frequency** components often occur as very short bursts, discontinuities, or transient phenomena in real signals: a **shorter time window** is therefore **required** to properly capture the instant at which they occur
- With the **STFT**, the length of the window is fixed and therefore fixing its length is always a **trade-off between poor frequency resolution at low frequencies and poor time resolution at high frequencies**

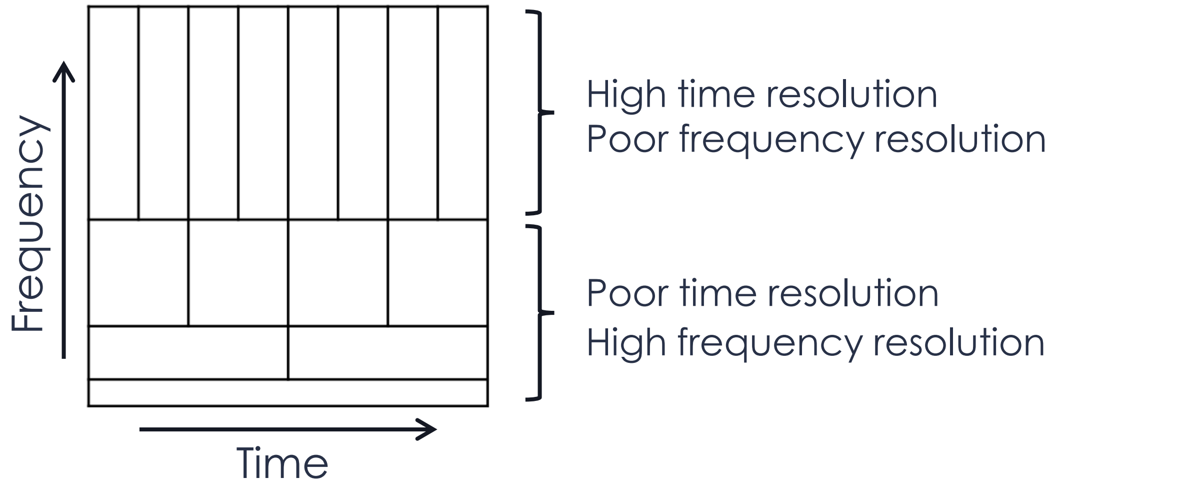


- ➢ The window of length L_1 is well suited for the LF component, but affects the time resolution of the HF component
- ➢ The window of length L_2 is well suited for the HF component, but affects the frequency resolution of the LF component

Wavelet analysis

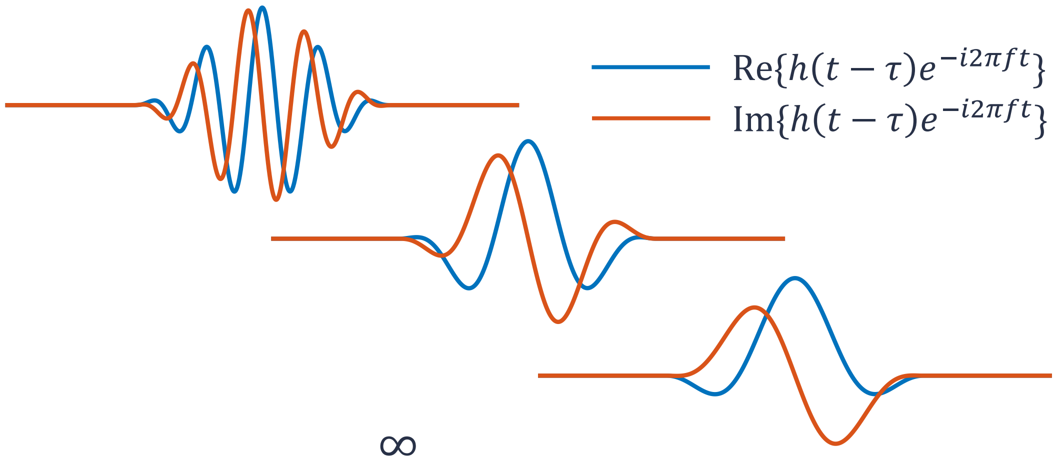
- The **STFT** projects the signal onto an orthogonal basis of **windowed sinusoids**: the **length of the window is the same** for all frequencies
- **Wavelet analysis** projects the signal onto a **more general orthogonal basis** of functions with **limited time support** (no prior windowing required as in the STFT) **inversely related to frequency**, and therefore often better suited to decompose the signal





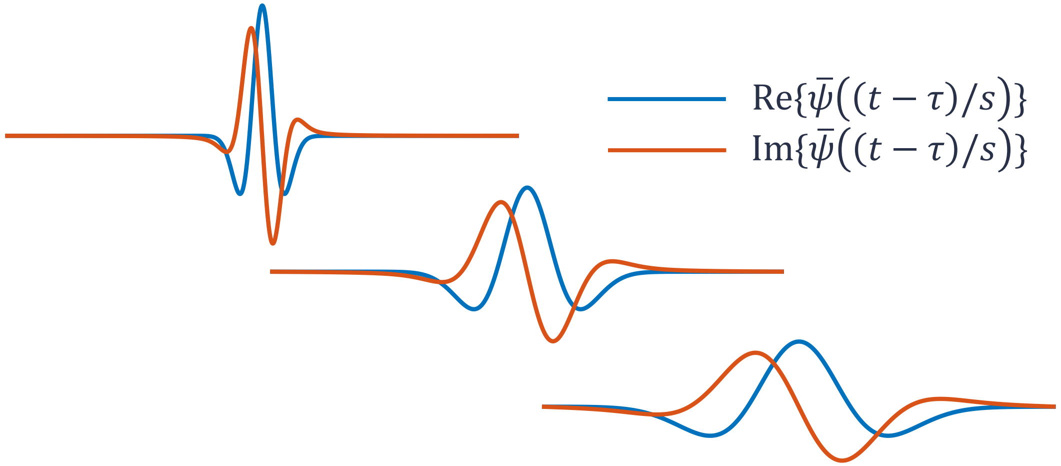
Wavelet analysis

STFT



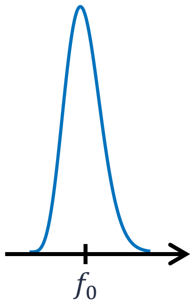
$$X_h(\tau, f) = \int_{-\infty}^{\infty} x(t)h(t-\tau)e^{-i2\pi ft}dt$$

Wavelet transform



$$X_\psi(\tau, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} x(t)\bar{\psi}\left(\frac{t-\tau}{s}\right)dt$$

- The wavelet transform is a **time-scale** (τ, s) rather than a time-frequency (τ, f) representation, but **can be used for time-frequency analysis** with $f = f_0/s$, where f_0 is the central frequency of the wavelet in the frequency domain



Wavelet analysis

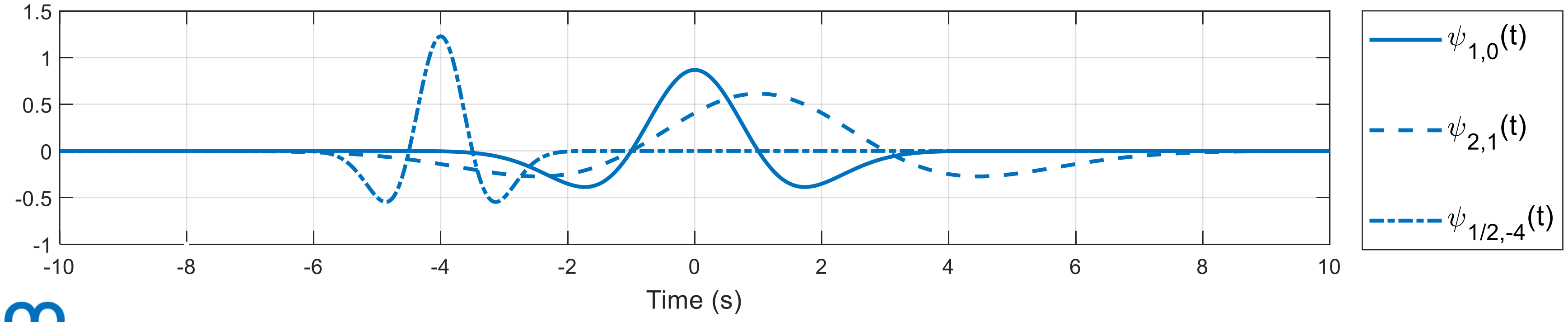
- Formally speaking, the **wavelet transform** projects the signal $x(t)$ onto a set of **zero-mean oscillatory functions** (*wavelets* $\psi_{s,\tau}(\cdot)$), created from a basic function (*mother wavelet* $\psi(\cdot)$) by translations and dilations

Example of the Mexican hat wavelet

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) = \frac{2}{\sqrt{3s}\pi^{1/4}}\left(1-\left(\frac{t-\tau}{s}\right)^2\right)e^{-\frac{1}{2}\left(\frac{t-\tau}{s}\right)^2}$$

- The parameter s is a **scale factor** ($s > 0$): if $s > 1$, the wavelet is dilated, and if $s < 1$, it is compressed

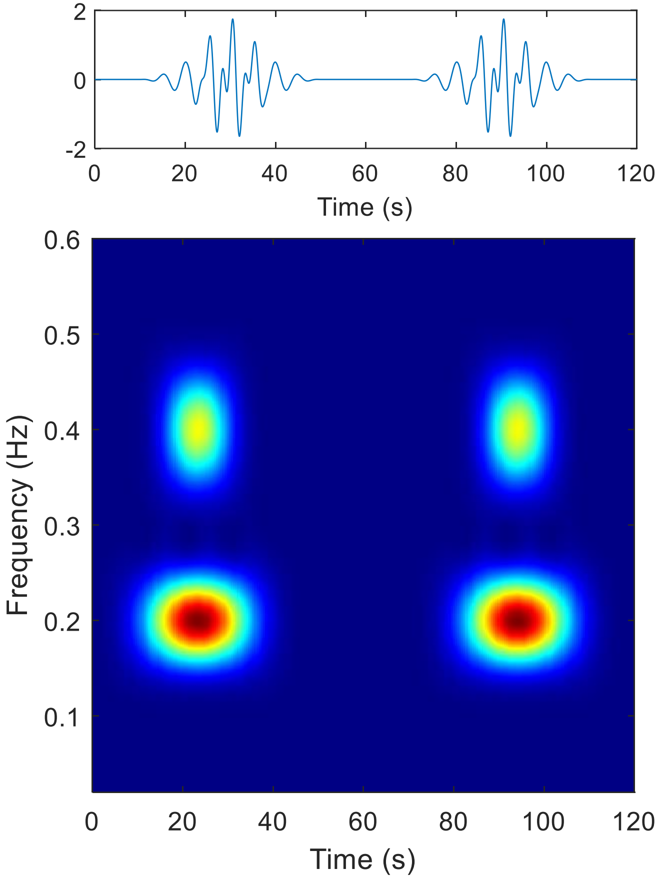
- The parameter τ is the time value around which the wavelet is centered



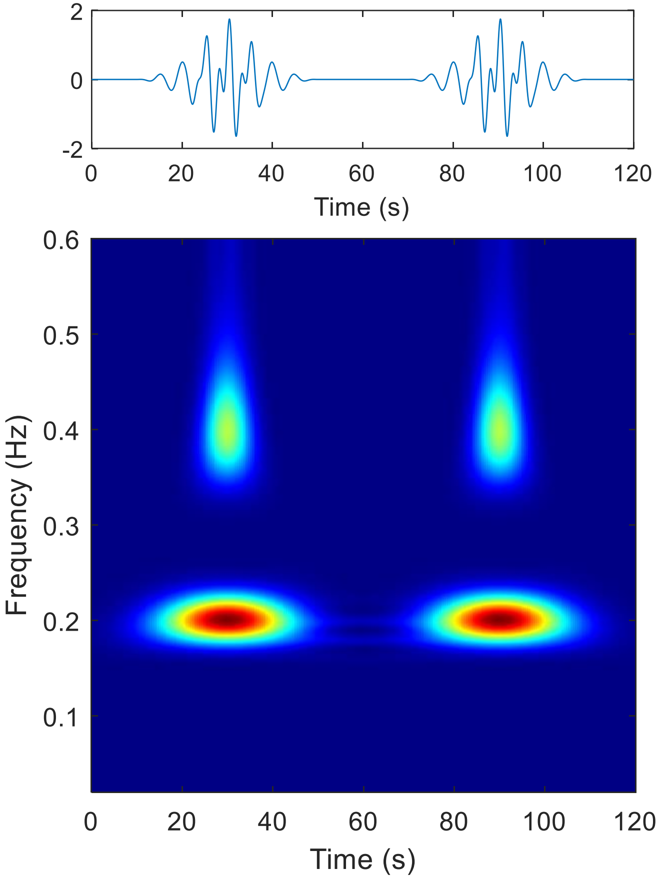
EPFL csem

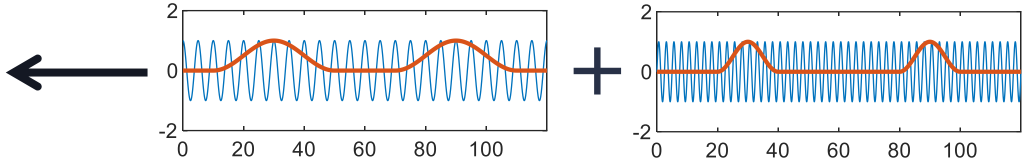
STFT vs. Wavelet analysis

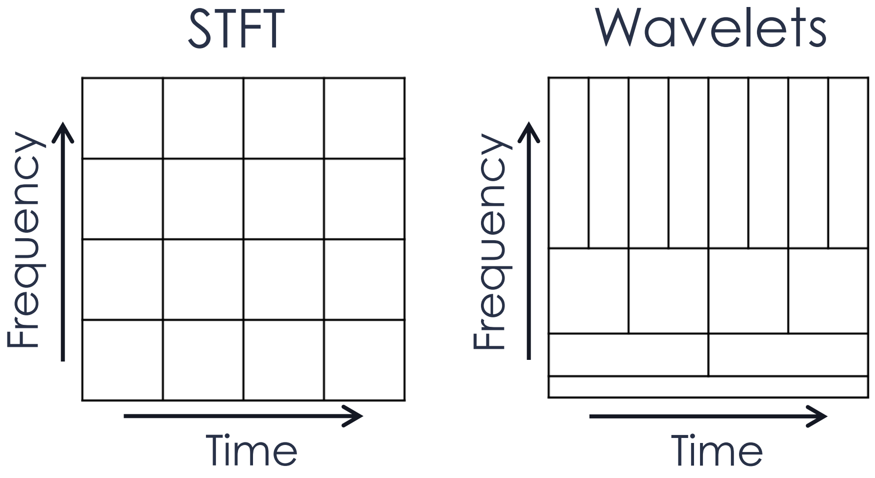
STFT



Wavelets



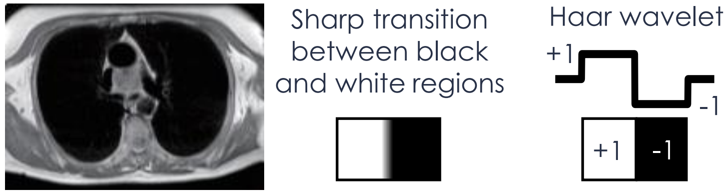


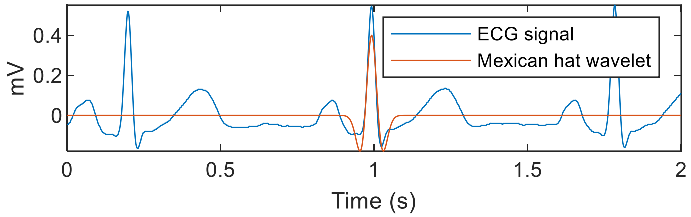


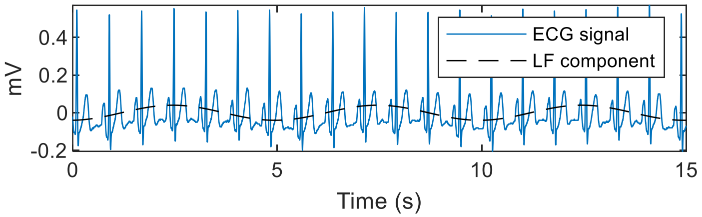
EPFL csem

STFT vs. Wavelet analysis: In practice, which to choose?

- **Wavelets** provide a more **flexible framework** (choice of basis function) and are generally computationally more efficient
- In many applications – particularly audio/image compression/processing, but also in the biomedical field (electroencephalography, electrocardiography, etc.) – wavelets can better **capture the morphologies/patterns** of some signals, and better cope with the **trade-off between time and frequency resolution**







- However, in practice, the **simplicity of use and interpretation of the STFT**, and the fact that it produces very similar results as wavelet analysis in the vast majority of cases, makes it a **perfectly reasonable default choice for time-frequency analysis**

References

- M. Akay, Ed., Time-Frequency and Wavelets in Biomedical Signal Processing, IEEE Press, Piscataway, NJ, 1998.
- B. Boashash, Ed., Time-Frequency Signal Analysis, Wiley, NY, 1992.
- S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, London, 1998.

- And for other sources, check **Moodle FAQ**