Hilbert Transform

in general:

$$s_a(t) = s(t) + i\mathcal{H}[s](t)$$

$$\mathcal{H}[s](t) = \frac{1}{\pi} P.V. \int_{-\infty}^{\infty} \frac{s(\tau)}{t - \tau} \, d\tau$$

Hilbert Transform

in general:

$$s_a(t) = s(t) + i\mathcal{H}[s](t)$$

$$\mathcal{H}[s](t) = \frac{1}{\pi} P.V. \int_{-\infty}^{\infty} \frac{s(\tau)}{t-\tau}\, d\tau$$

$$s_a(t) = R(t)e^{i\phi(t)}$$

R: instantaneous amplitude

ϕ: instantaneous phase

Hilbert Transform

in general:

$$s_a(t) = s(t) + i\mathcal{H}[s](t)$$

$$\mathcal{H}[s](t) = \frac{1}{\pi} P.V. \int_{-\infty}^{\infty} \frac{s(\tau)}{t-\tau}\, d\tau$$

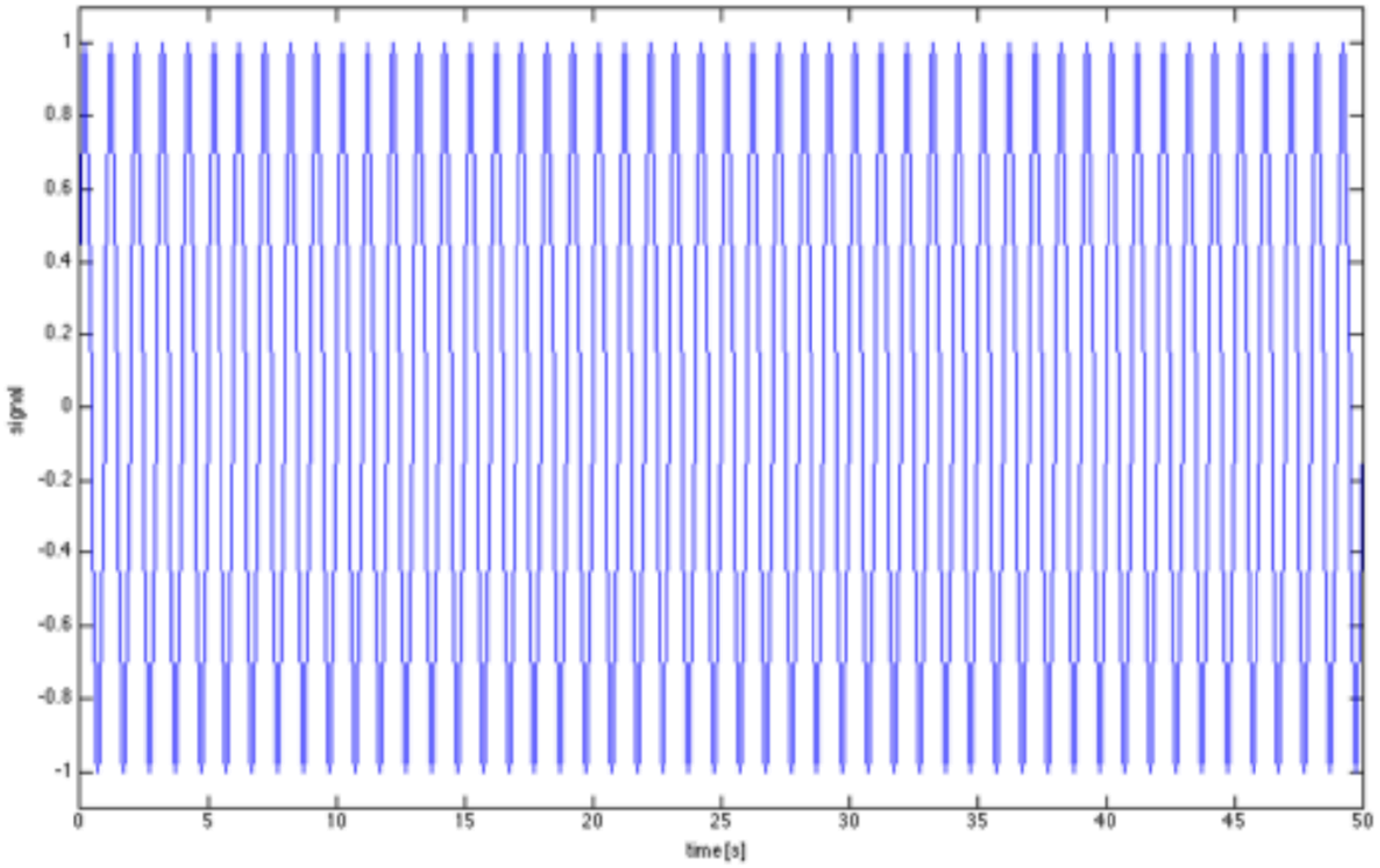
$$s_a(t) = R(t)e^{i\phi(t)}$$

R: instantaneous amplitude

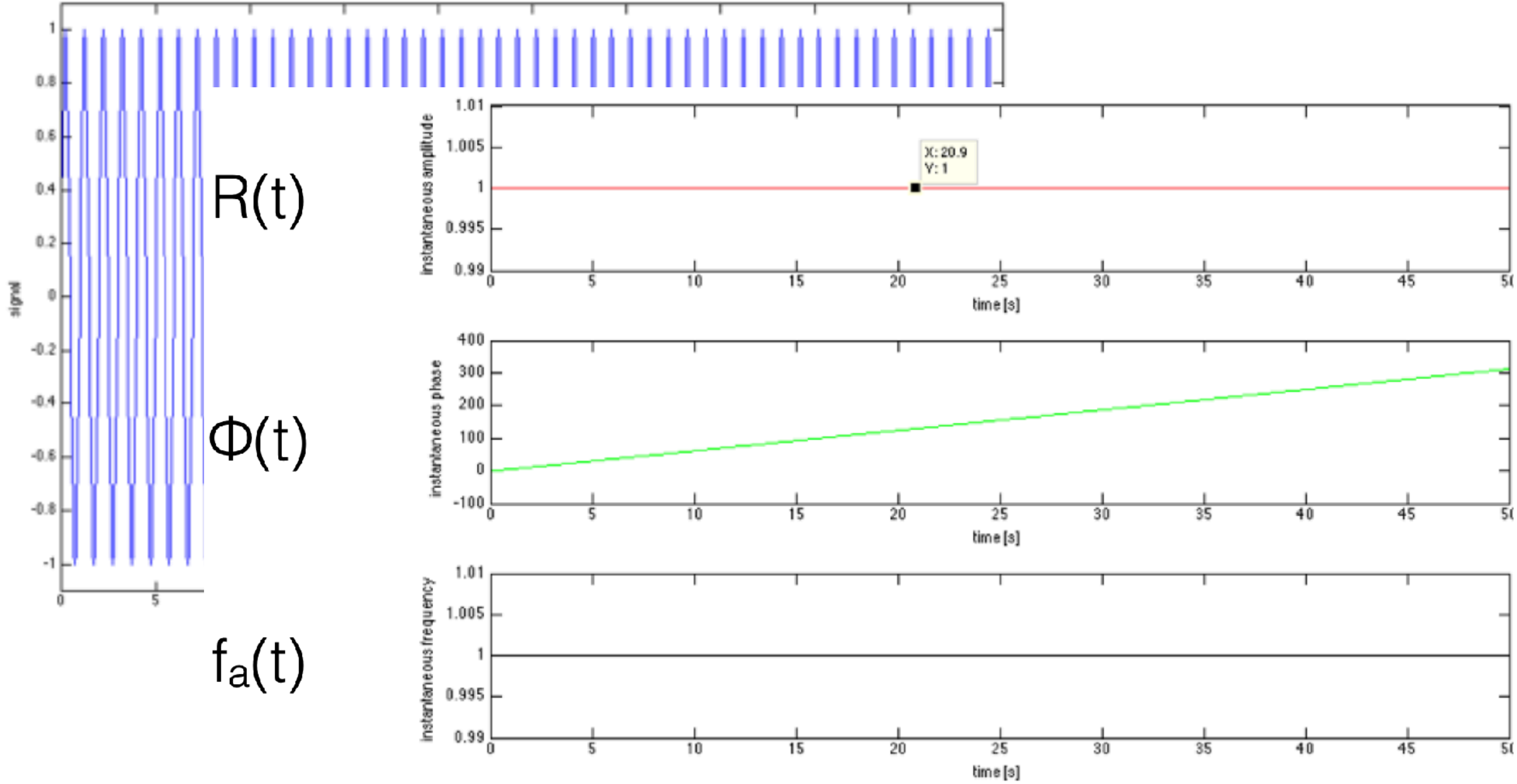
ϕ: instantaneous phase

→ instantaneous frequency $f_a(t) = \frac{1}{2\pi}\frac{d\phi(t)}{dt}$

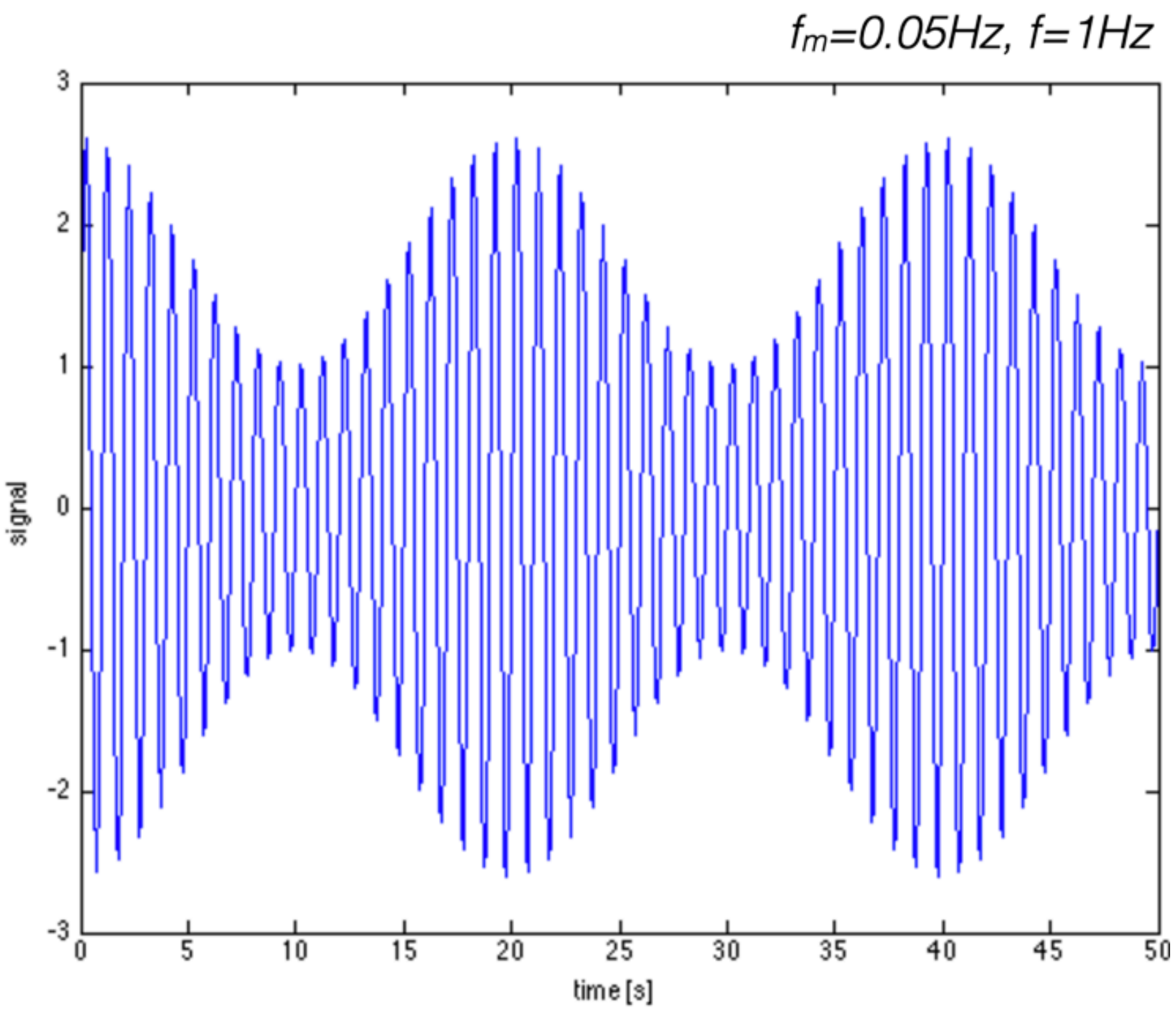
example signal: single frequency oscillation $s(t)=\sin(2\pi t)$



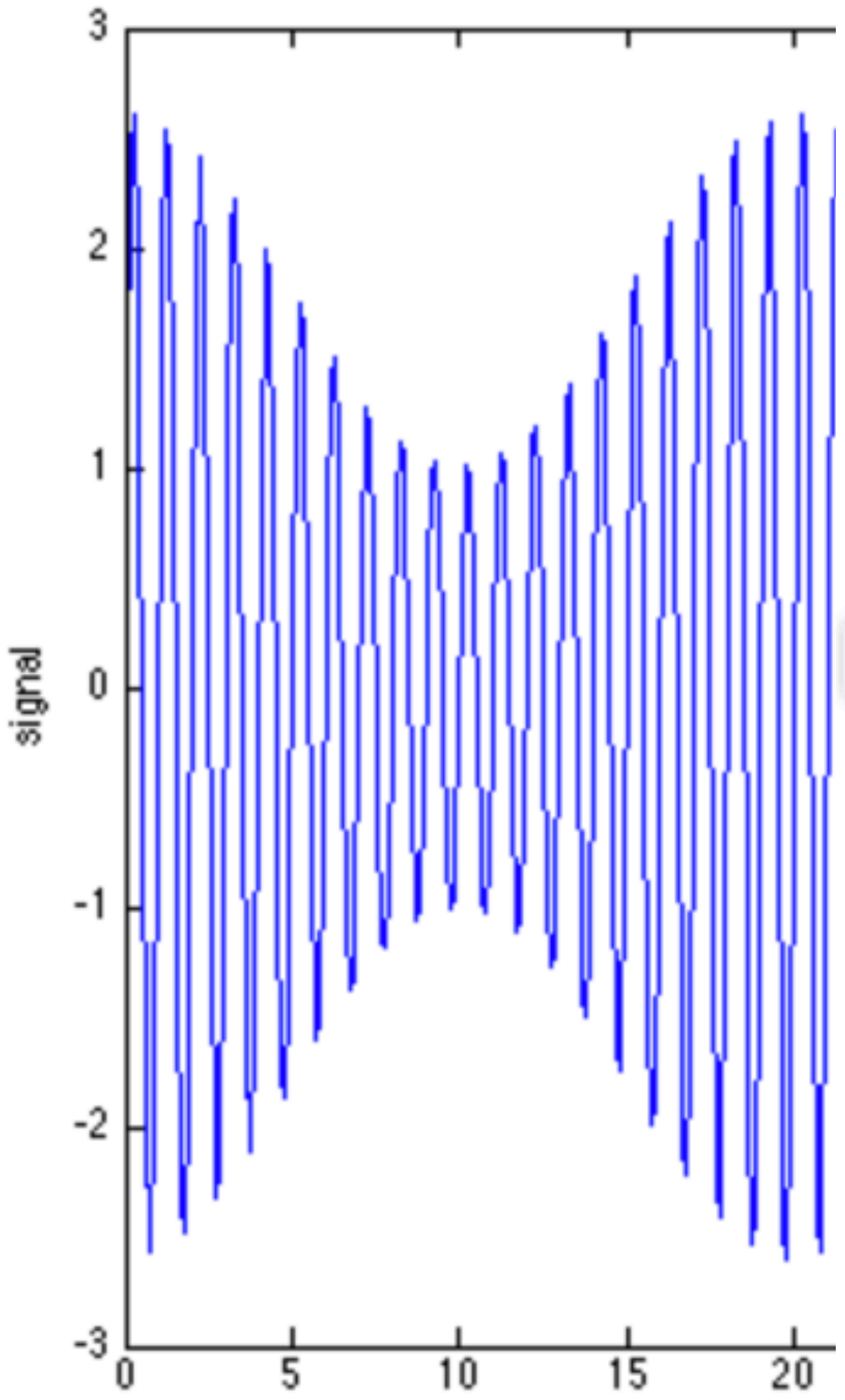
example signal: single frequency oscillation $s(t)=\sin(2\pi t)$



example: amplitude-modulated oscillation $s(t) = [1 + 0.8\cos(2\pi f_m t)]\sin(2\pi f t)$



$$s(t) = [1 + 0.8\cos(2\pi f_m t)]\sin(2\pi f t)$$

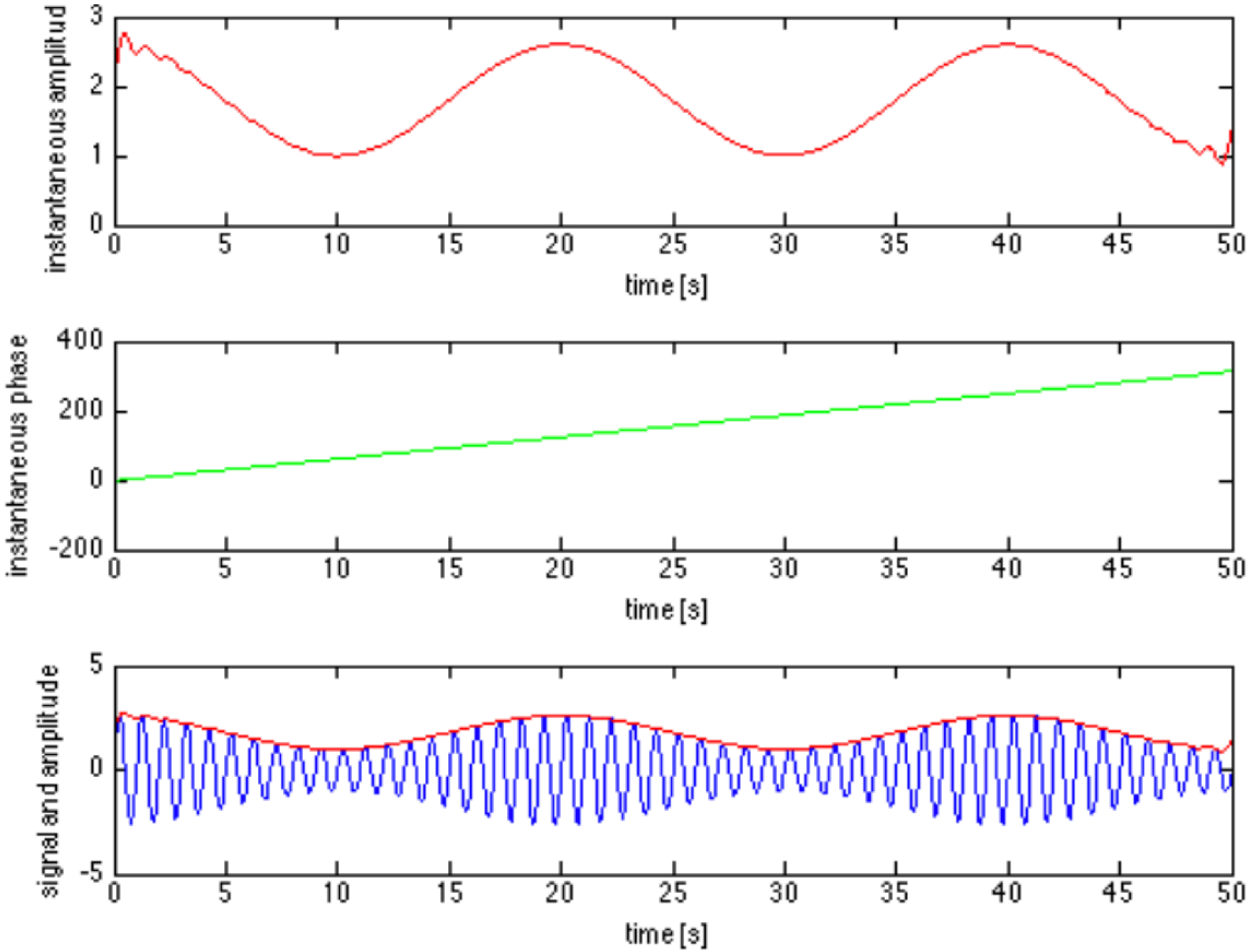


R(t)

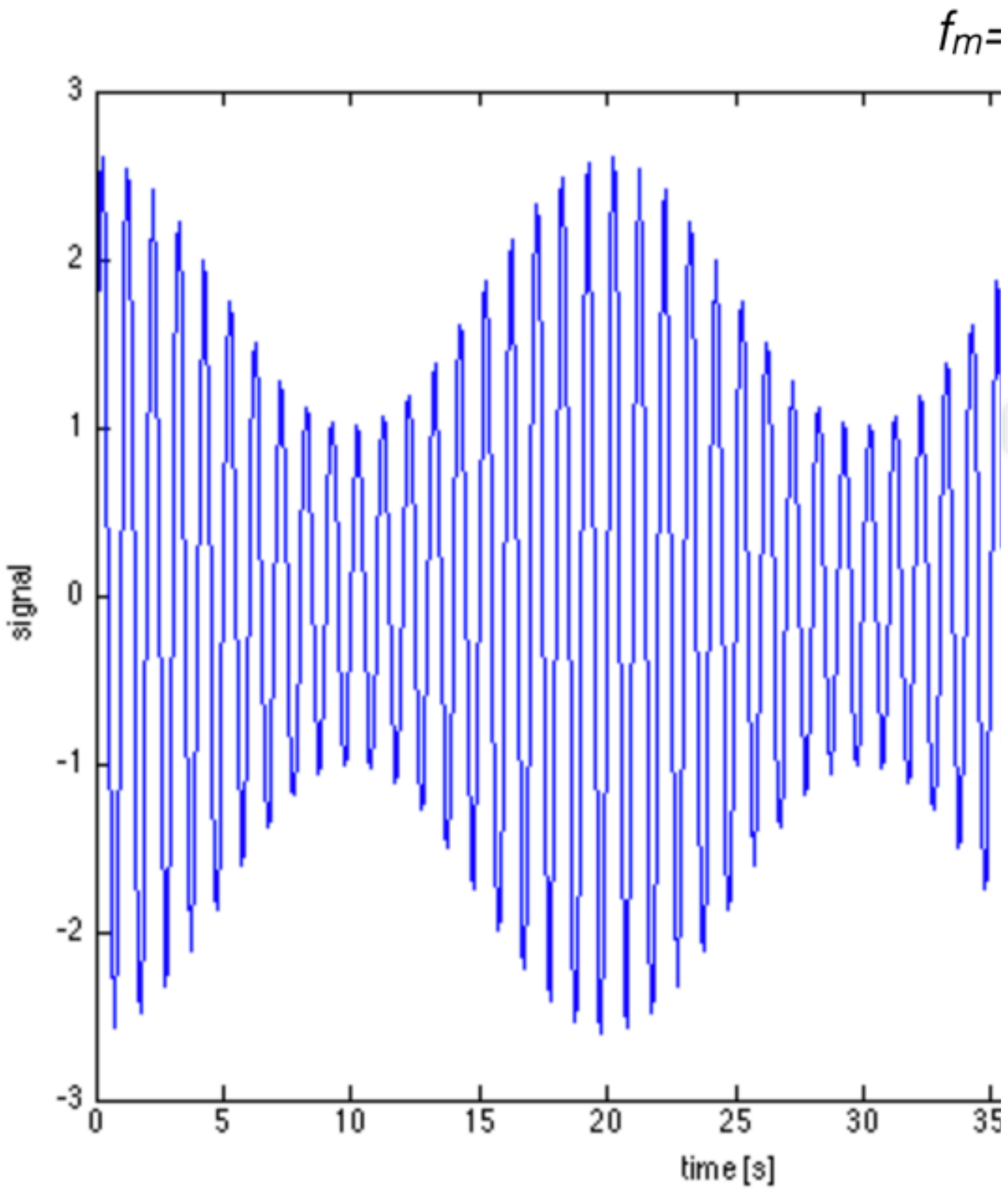
Screenshot

Φ(t)

R(t)cos(Φ(t))



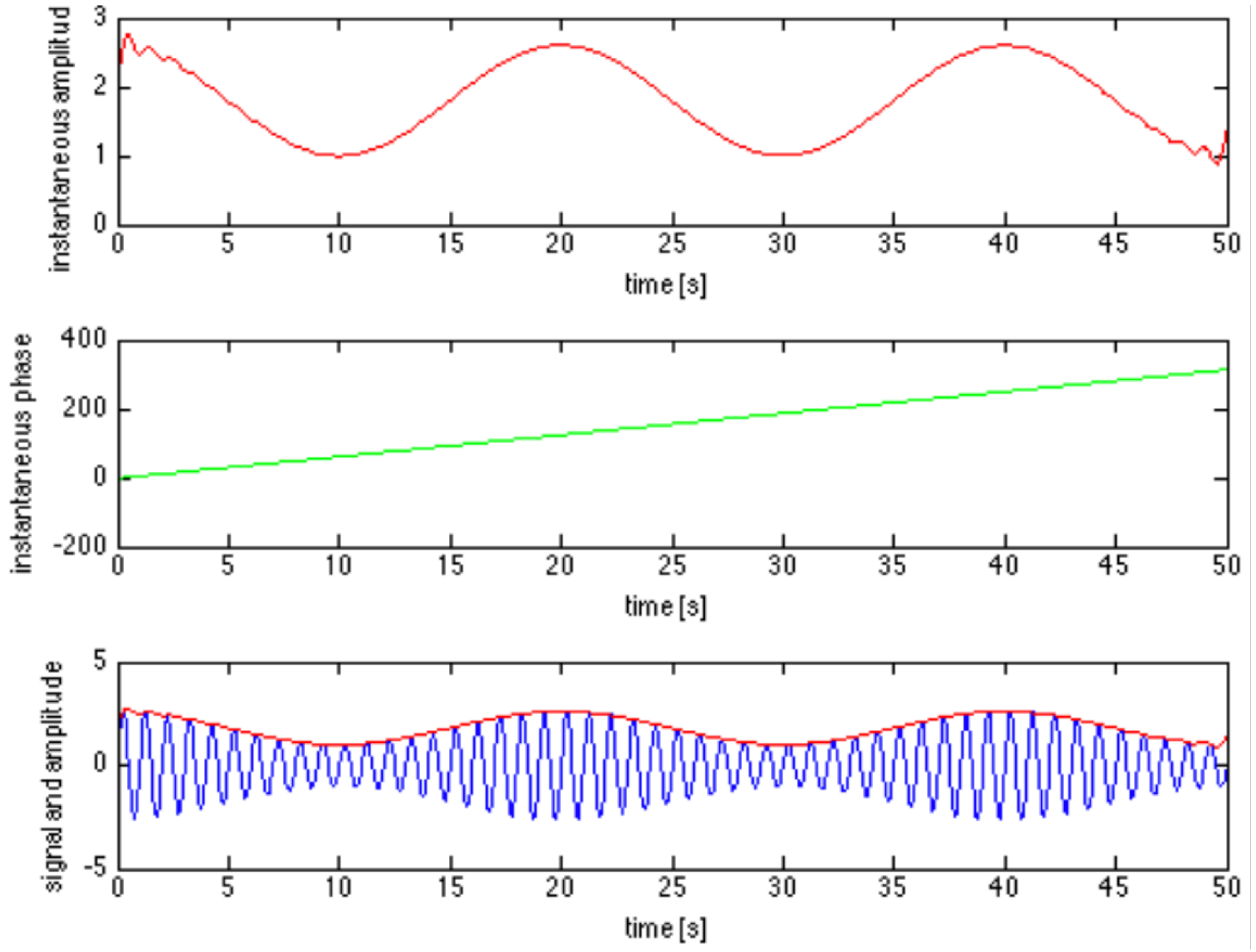
$$s(t) = [1 + 0.8\cos(2\pi f_m t)]\sin(2\pi f t)$$



R(t)

Φ(t)

R(t)cos(Φ(t))



Hilbert transform allows to determine the amplitude modulation

$$s(t) = [1 + 0.8\cos(2\pi f_m t)]\sin(2\pi f t)$$

$$= \sin(2\pi f t) + 0.4\sin(2\pi(f + f_m)t) + 0.4\sin(2\pi(f - f_m)t)$$

$$s(t) = [1 + 0.8\cos(2\pi f_m t)]\sin(2\pi f t)$$
$$= \sin(2\pi f t) + 0.4\sin(2\pi(f + f_m)t) + 0.4\sin(2\pi(f - f_m)t)$$

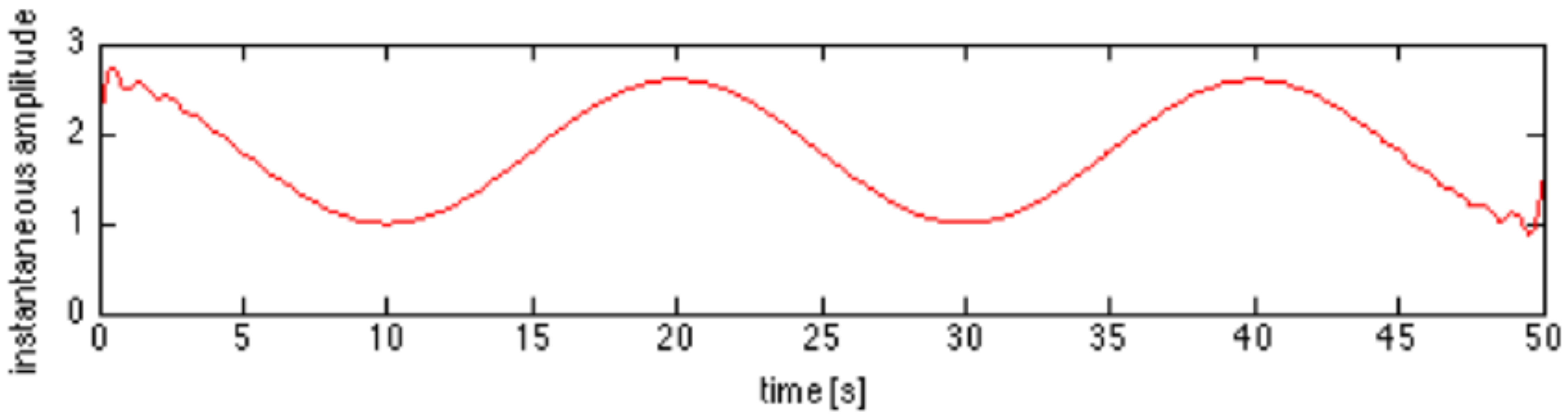
3 frequencies: $f \; , \; f + f_m \; , \; f - f_m$

$$s(t) = [1 + 0.8\cos(2\pi f_m t)]\sin(2\pi f t)$$

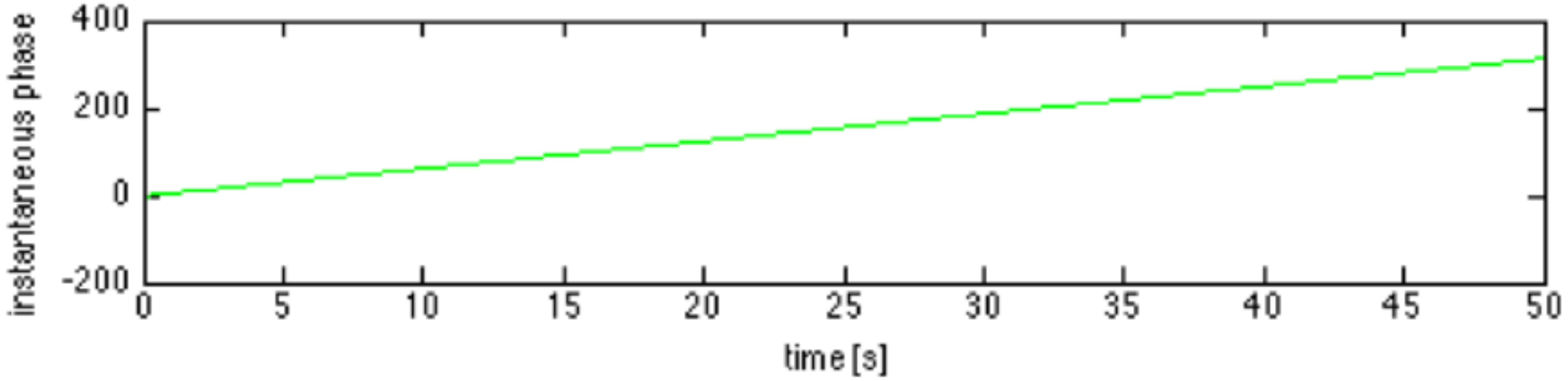
$$= \sin(2\pi f t) + 0.4\sin(2\pi (f + f_m) t) + 0.4\sin(2\pi (f - f_m) t)$$

3 frequencies: $f\ ,\ f + f_m\ ,\ f - f_m$

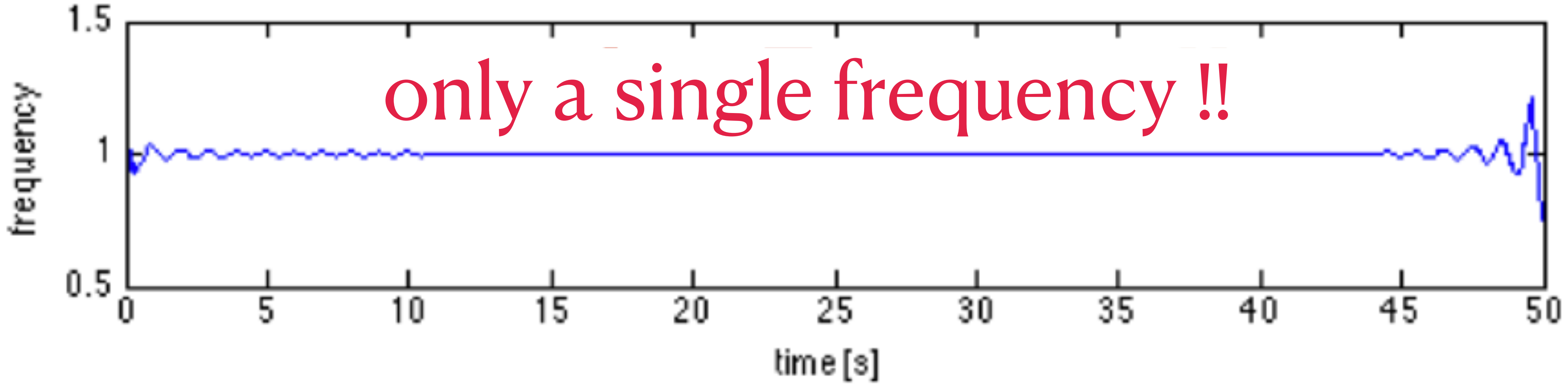
R(t)



Φ(t)



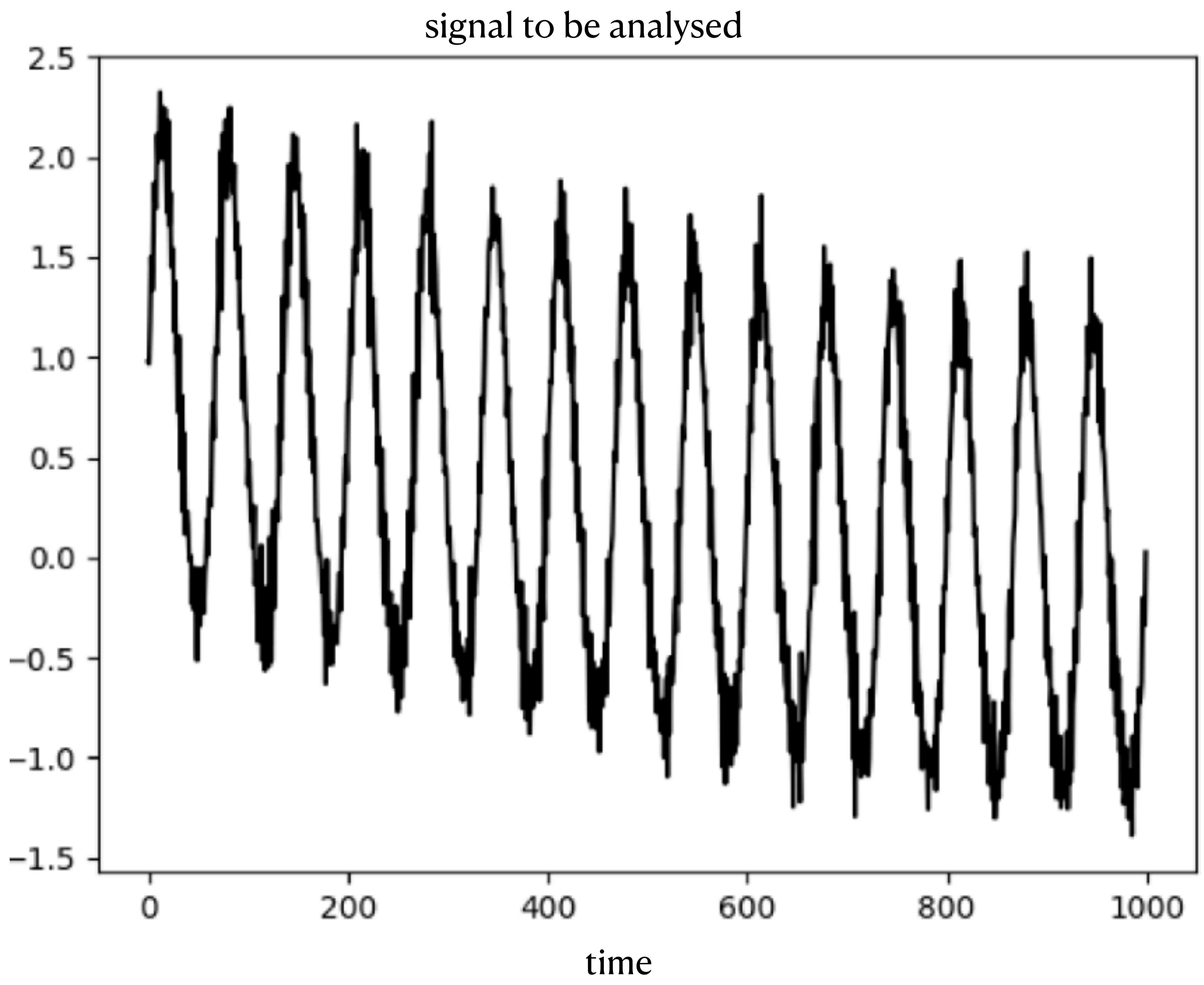
$f_a(t)$



analytical signal allows to describe amplitude modulation,

but poorly instantaneous frequency !!

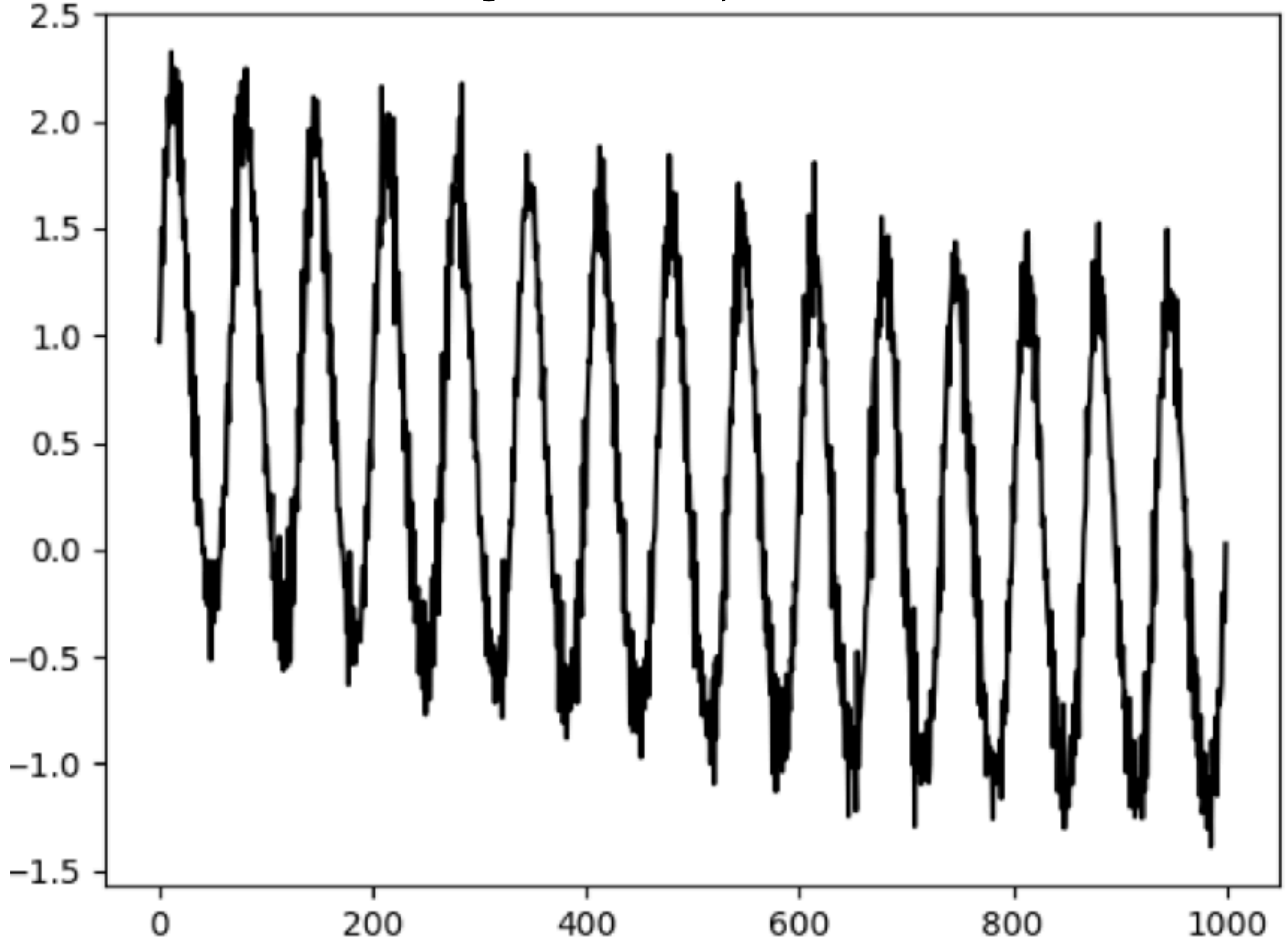
Empirical Mode Decomposition (EMD)



taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

Empirical Mode Decomposition (EMD)

signal to be analysed

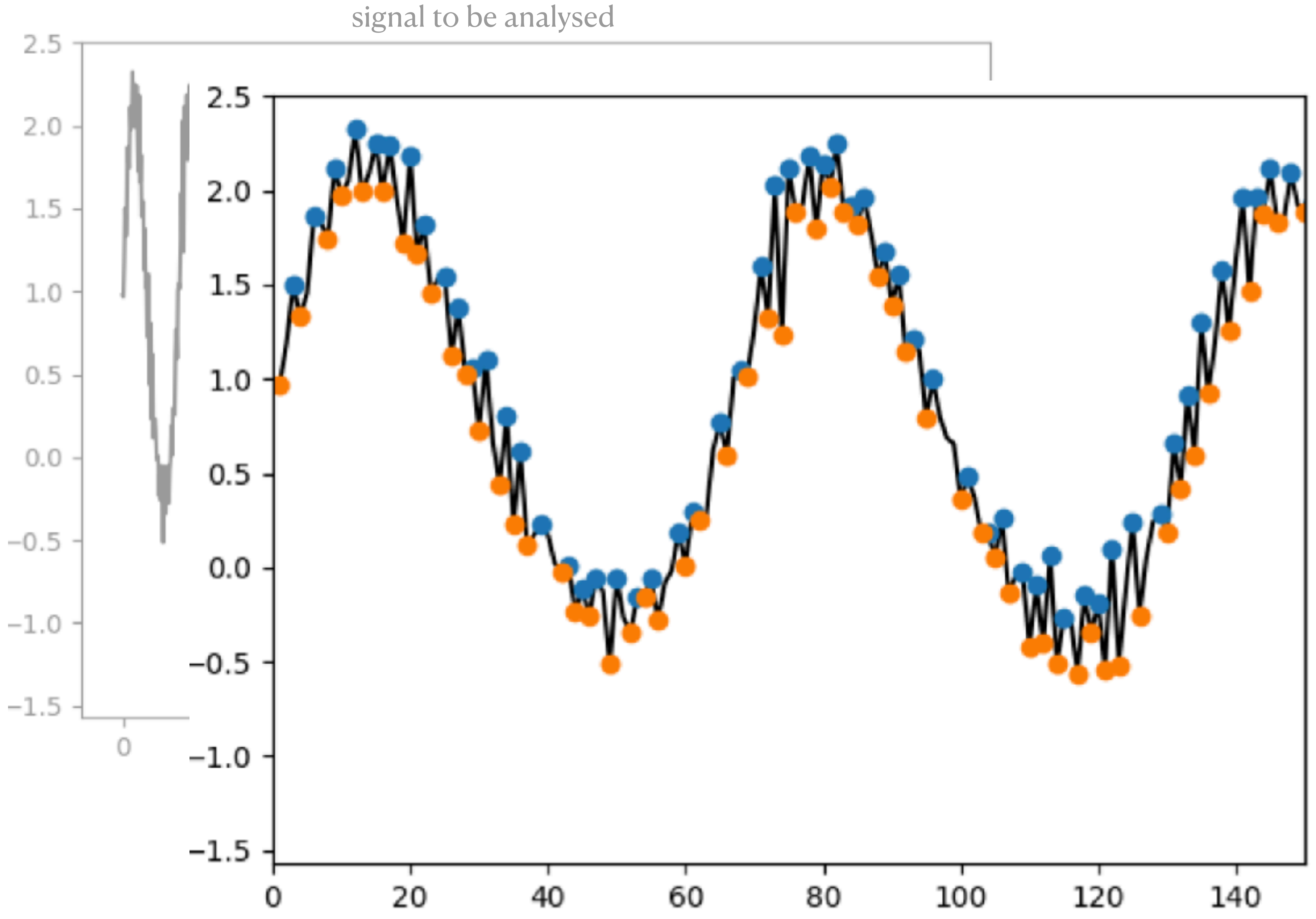


time

first analysis step : **sifting**

taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

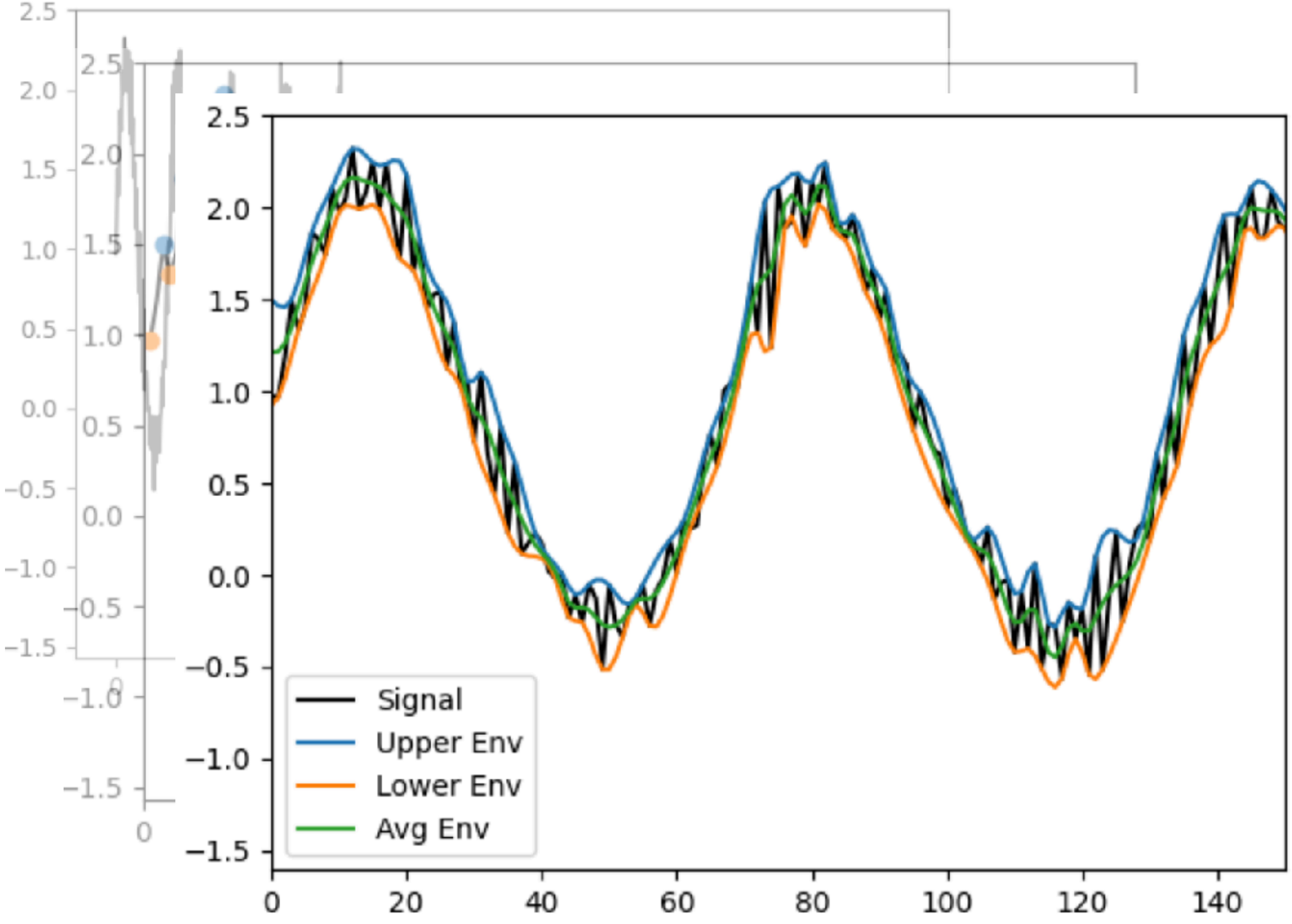
Empirical Mode Decomposition (EMD)



determine maxima and minima

taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

Empirical Mode Decomposition (EMD)



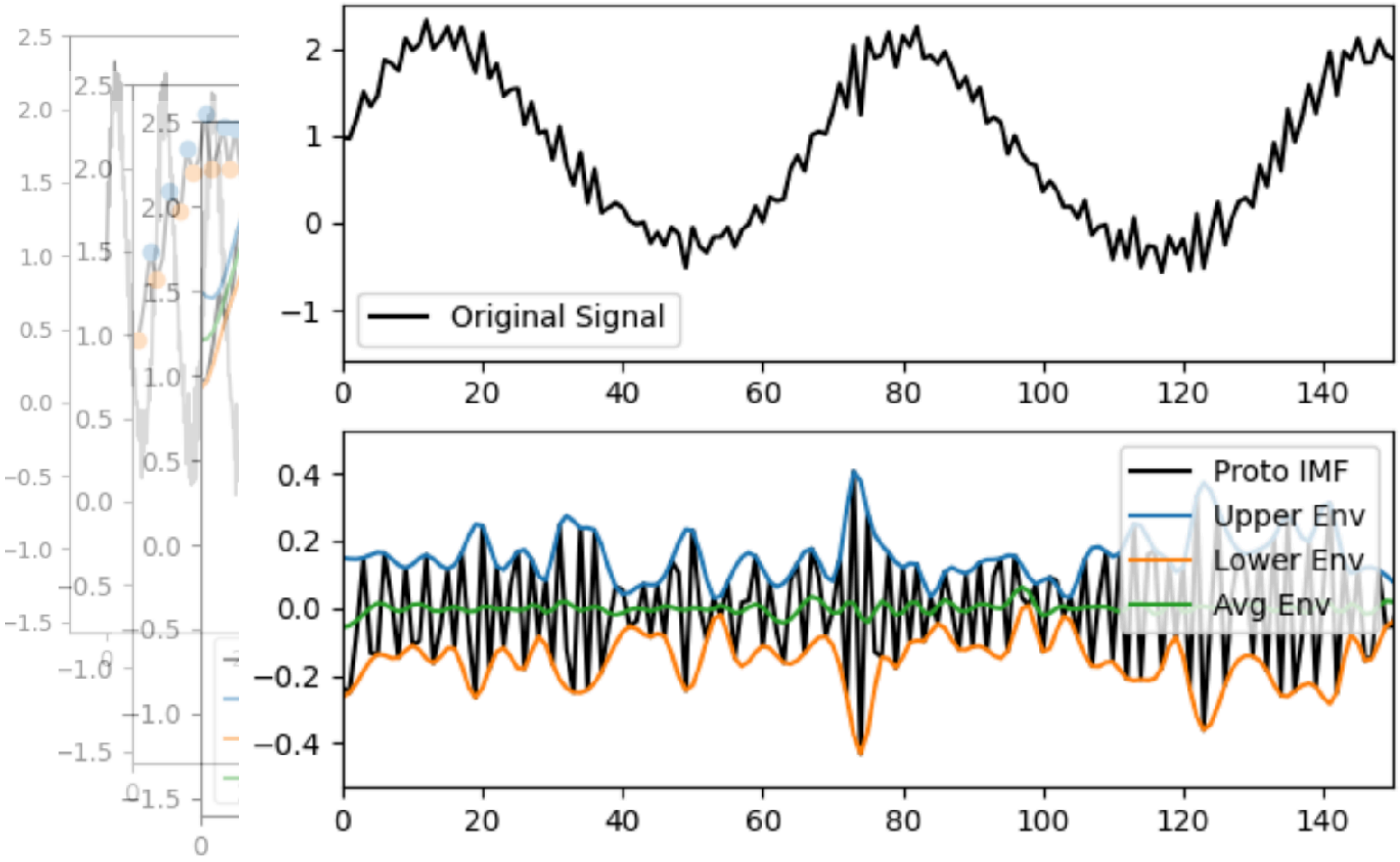
determine upper and lower envelope

and

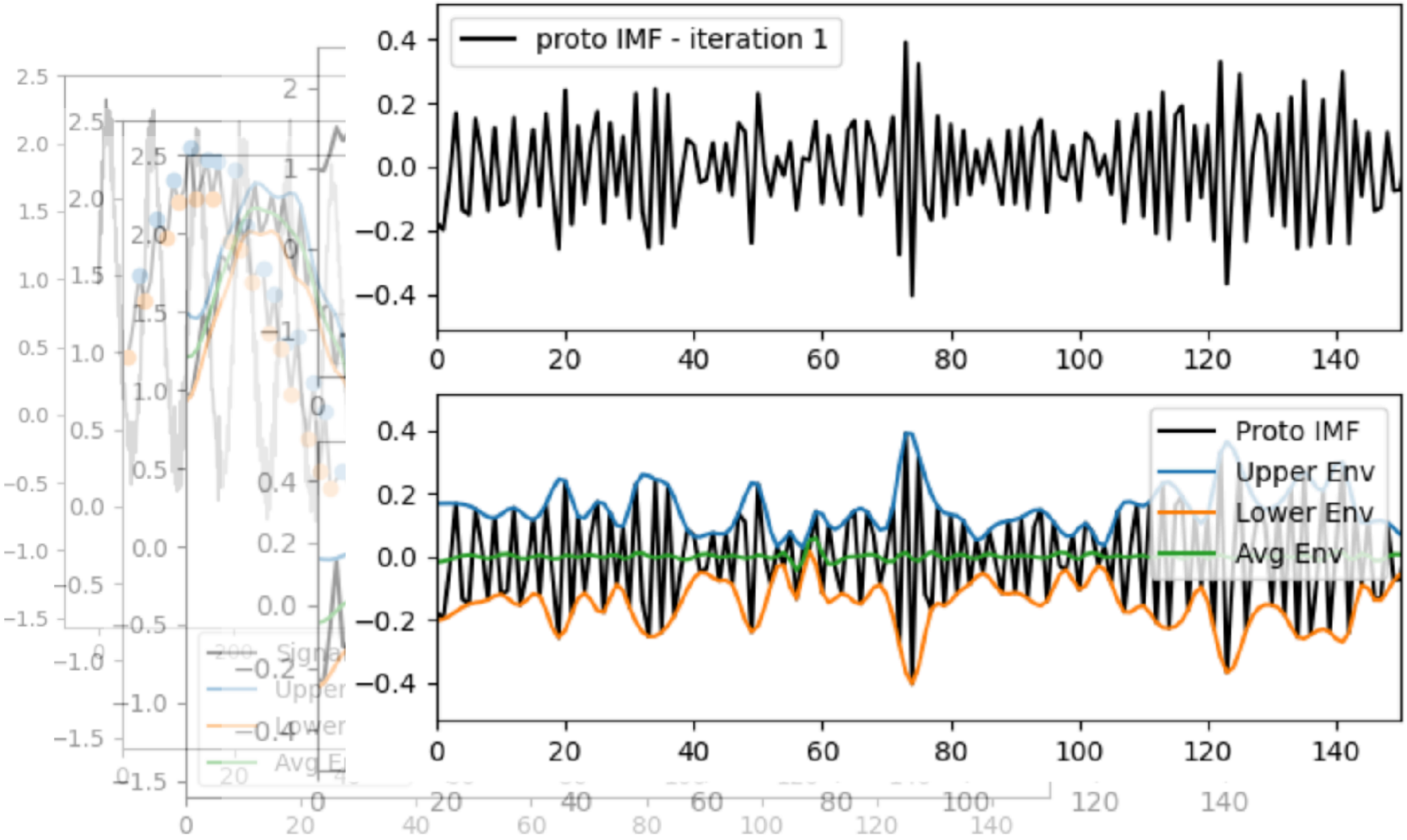
mean of maxima and minima envelope

taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

Empirical Mode Decomposition (EMD)

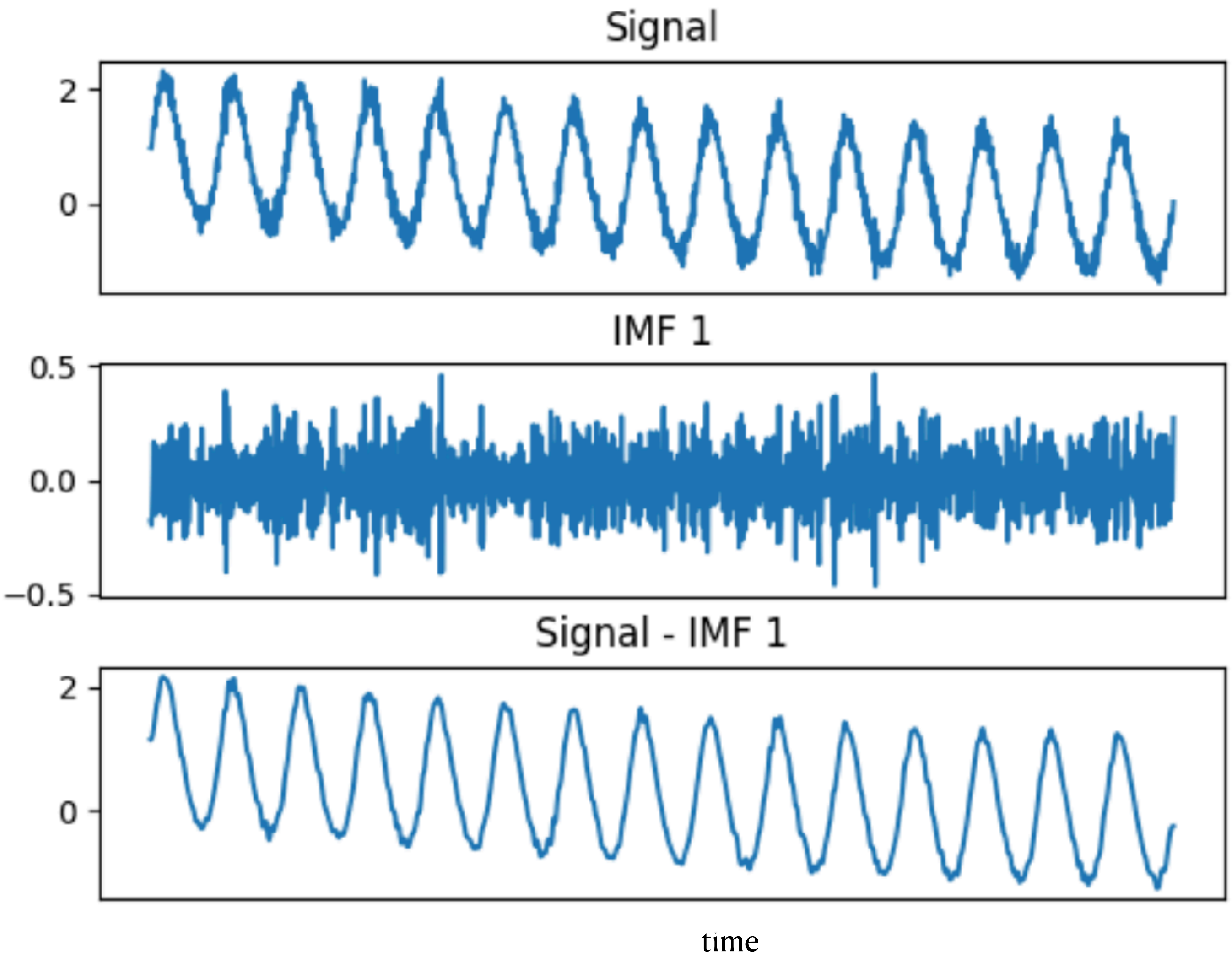


subtract envelope mean from signal



difference signal is first **Intrinsic Mode Function (IMF)**

taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html



next sifting step yields next IMF

taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

- Each $IMF_k(t)$ is a time series and a component of the signal s(t)

$$s(t) = \sum_{k=1}^{N} \mathrm{IMF}_k(t) + r(t)$$

- Each $IMF_k(t)$ is a time series and a component of the signal s(t)

$$s(t) = \sum_{k=1}^{N} \mathrm{IMF}_k(t) + r(t)$$

- the residual r(t) is a time series that can not be reduced anymore to an IMF

- Each $IMF_k(t)$ is a time series and a component of the signal s(t)

$$s(t) = \sum_{k=1}^{N} \mathrm{IMF}_k(t) + r(t)$$

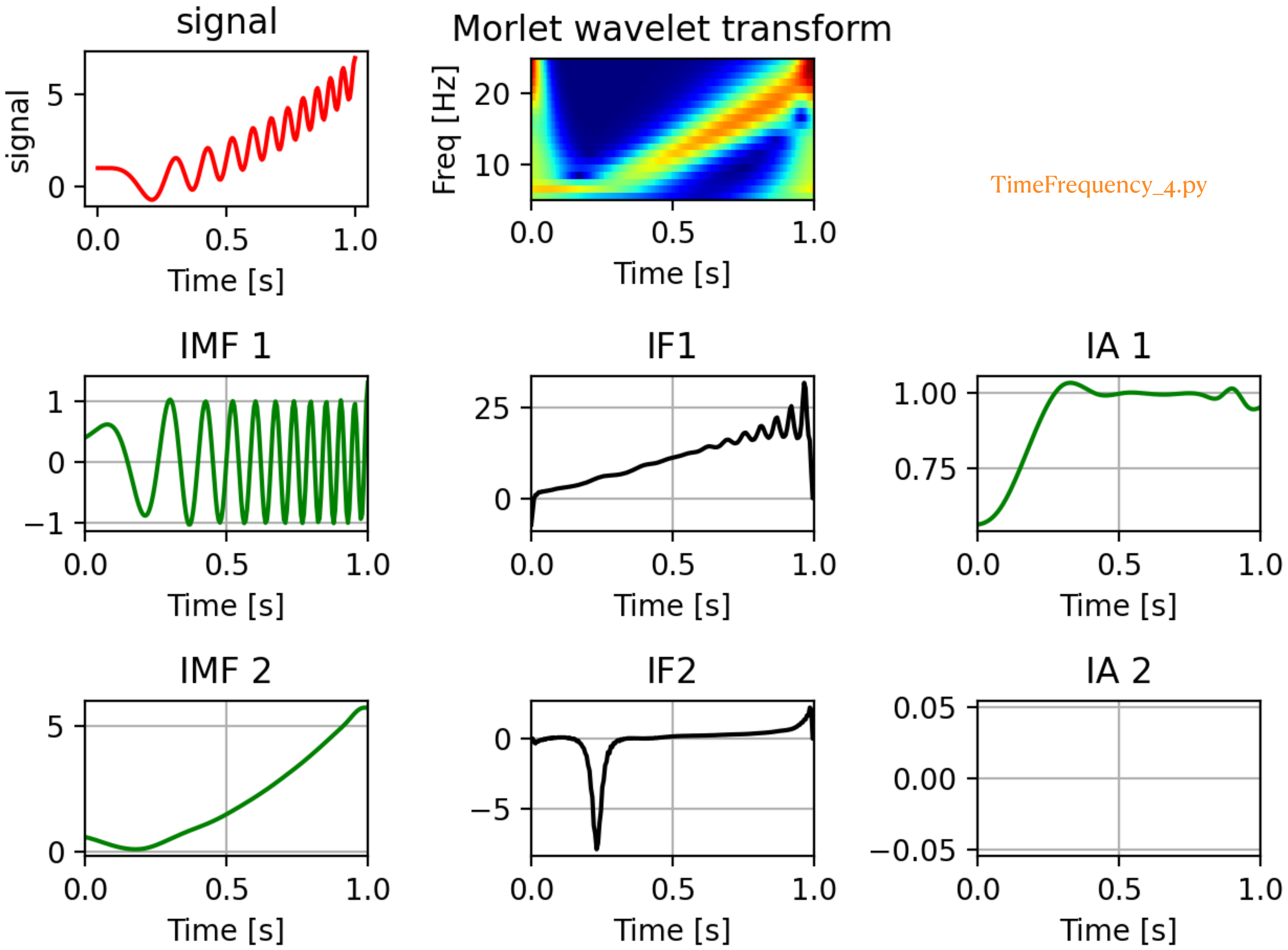
- the residual r(t) is a time series that can not be reduced anymore to an IMF

- when does sifting process end ?

  - ✤ fixed number of sifting iterations
  - ✤ threshold criterion

$$\sum_{n=1}^{T} \frac{|\mathrm{IMF}_{k-1}(t_n) - \mathrm{IMF}_k(t_n)|^2}{\mathrm{IMF}_{k-1}^2(t_n)} < \epsilon_k$$

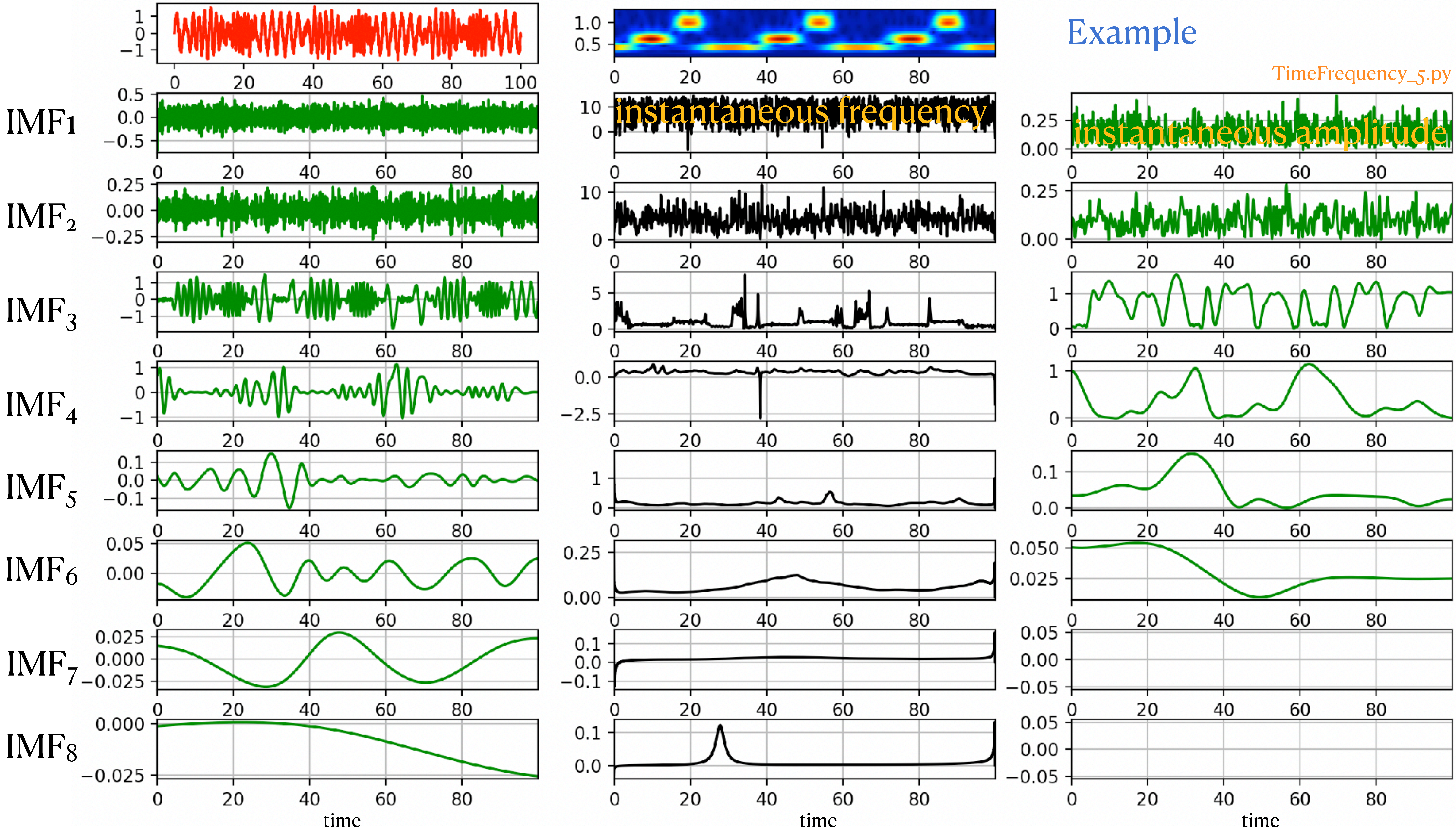
Example

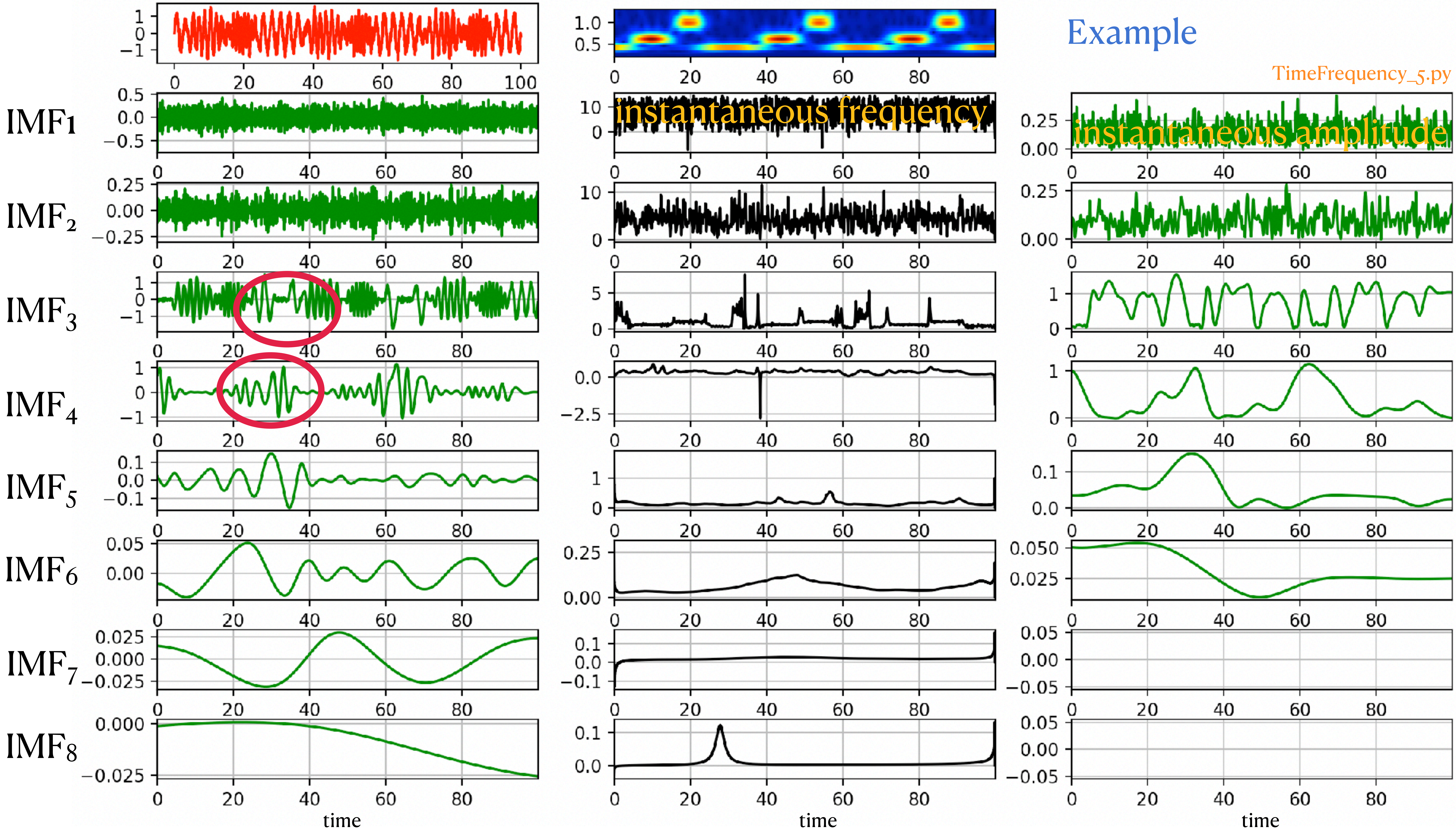


TimeFrequency_4.py

Example

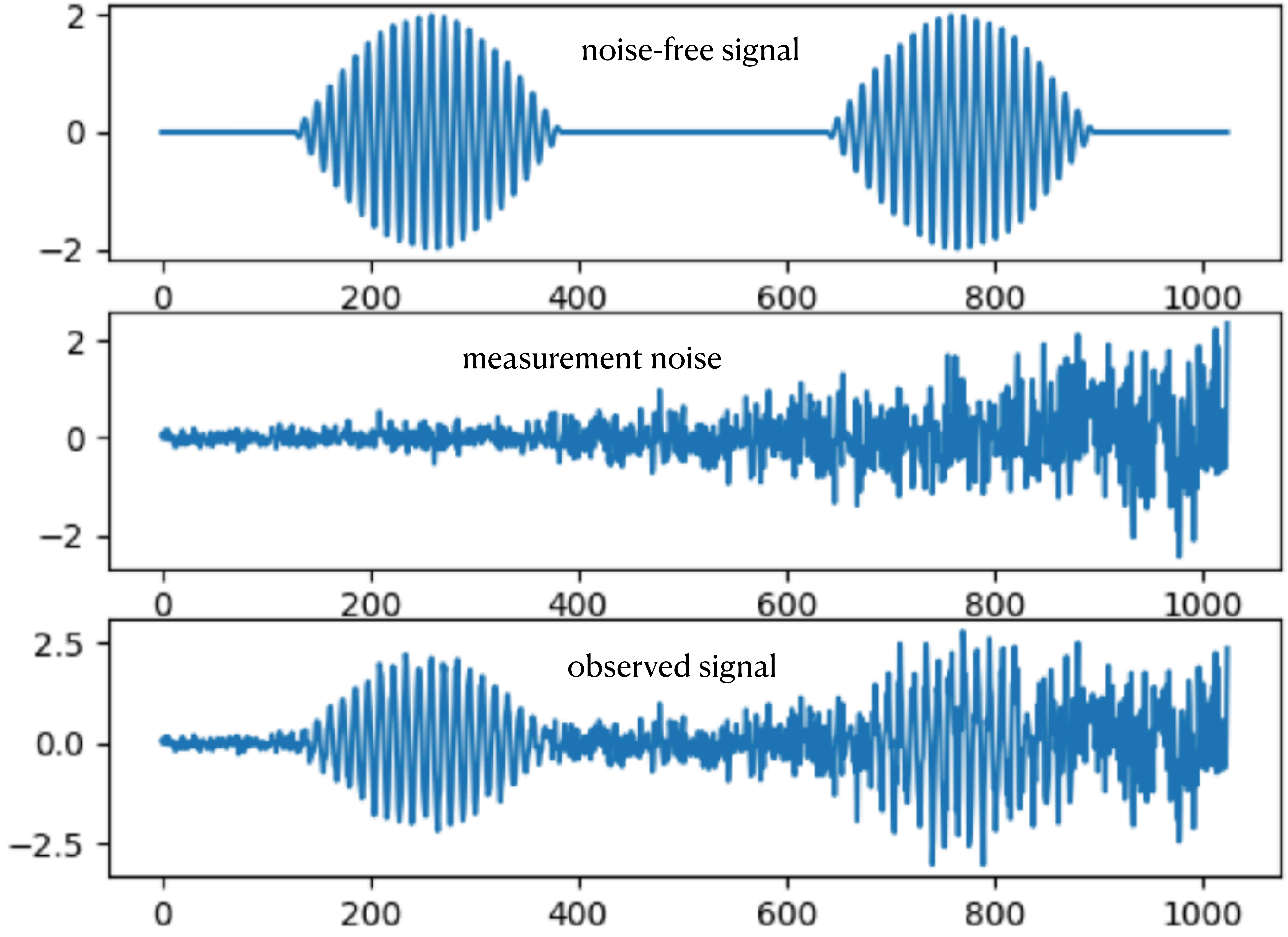
TimeFrequency_5.py





Example
TimeFrequency_5.py
instantaneous frequency
instantaneous amplitude
IMF1
IMF2
IMF3
IMF4
IMF5
IMF6
IMF7
IMF8
time
time
time

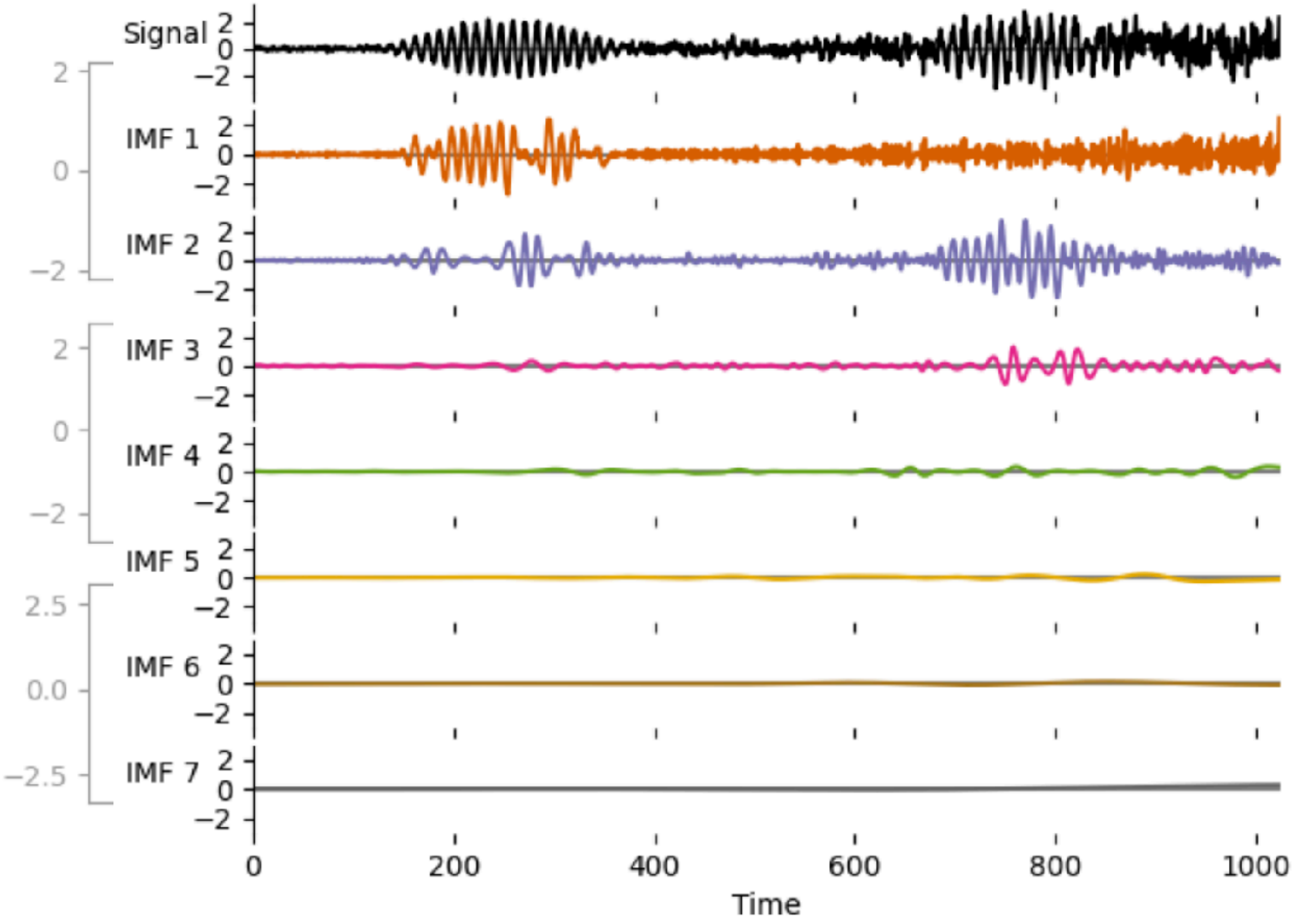
Ensemble Empirical Mode Decomposition (EEMD)



at first : application of EMD

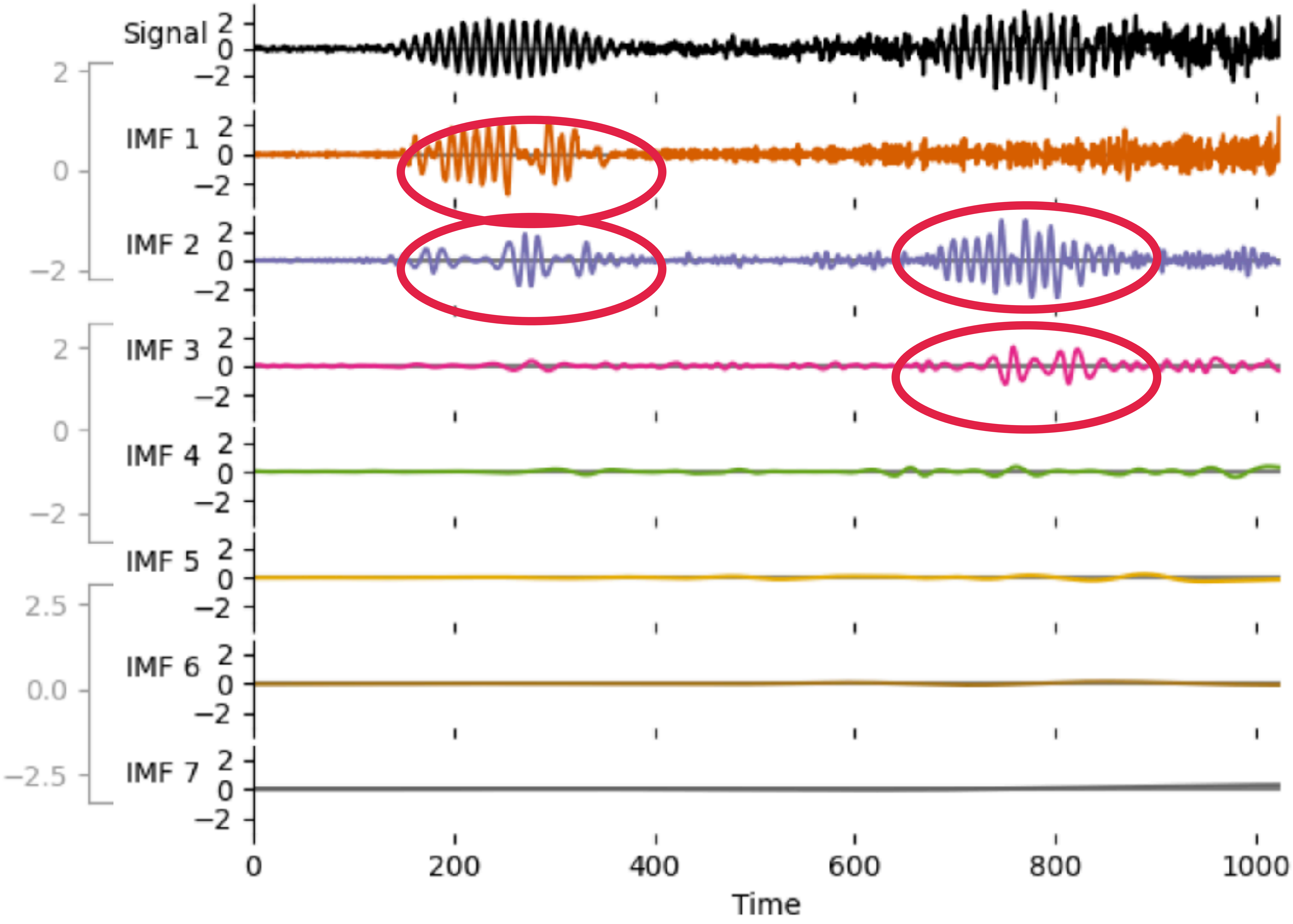
taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

Ensemble Empirical Mode Decomposition (EEMD)



taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

Ensemble Empirical Mode Decomposition (EEMD)



burst is distributed over several IMFs

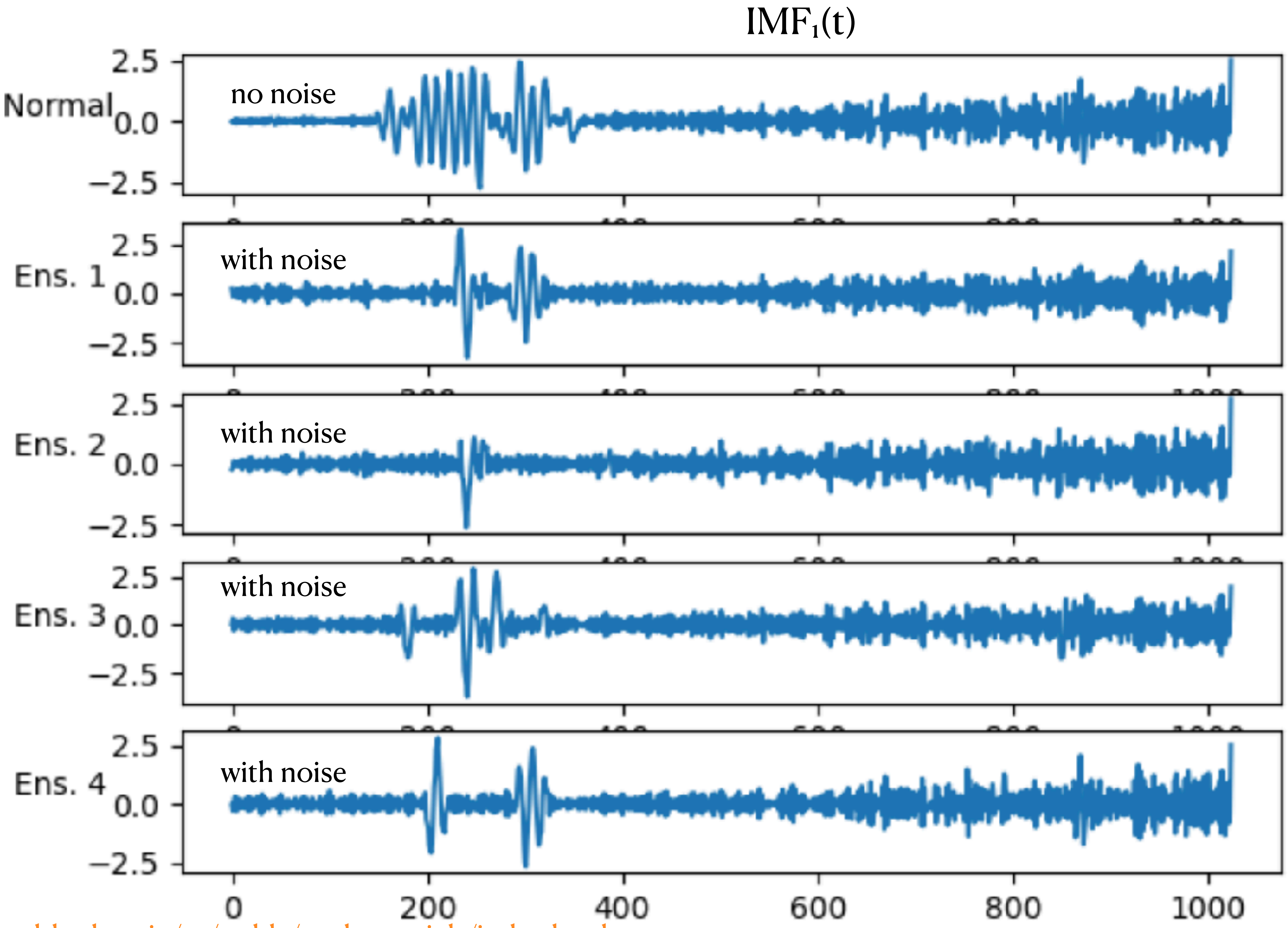
EMD does not separate bursts and noise

taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

Ensemble Empirical Mode Decomposition (EEMD)

idea:

- add noise to signal L times
  to create L ensemble members



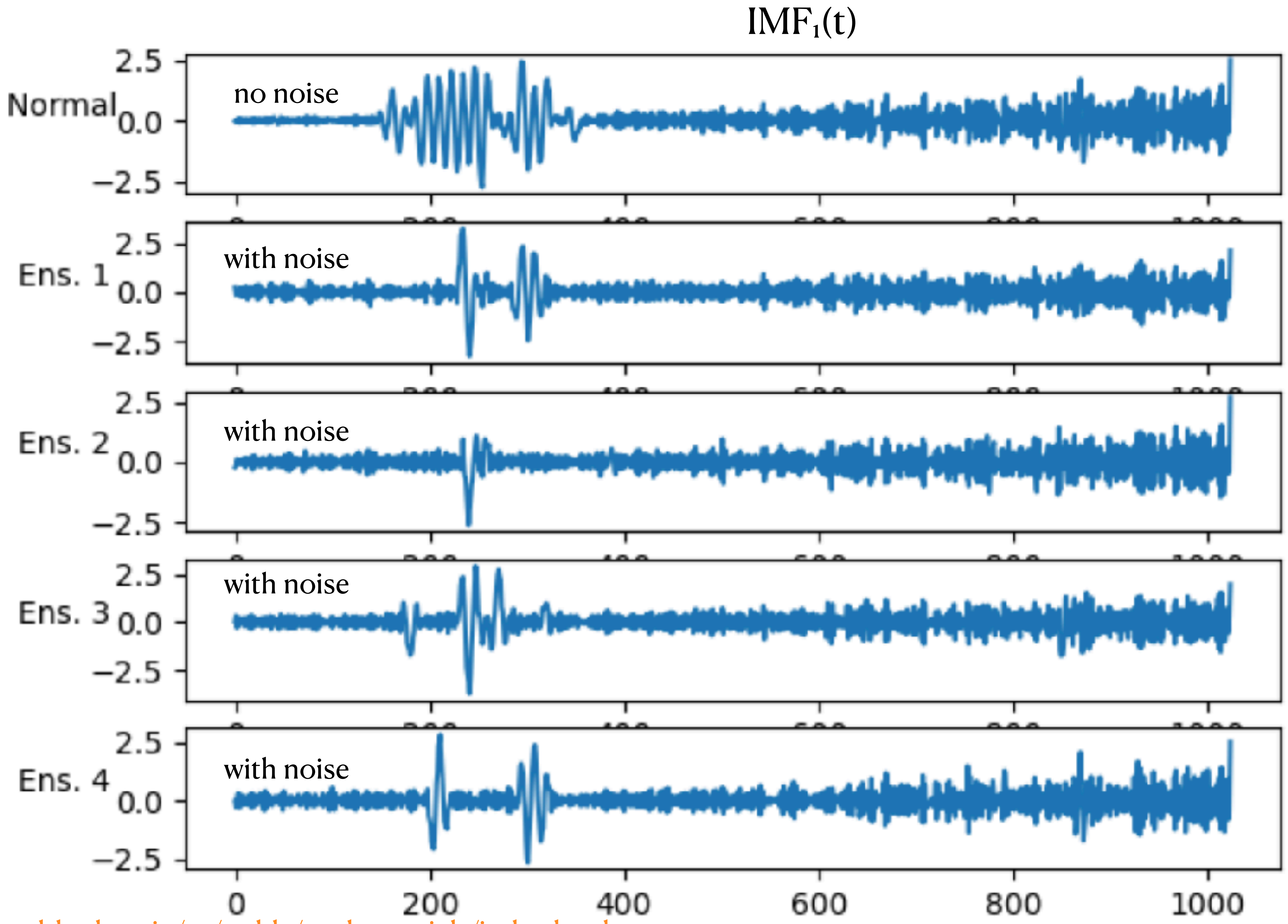
taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

Ensemble Empirical Mode Decomposition (EEMD)

idea:

- add noise to signal L times
  to create L ensemble members

- perform EMD on
  all ensemble members



taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

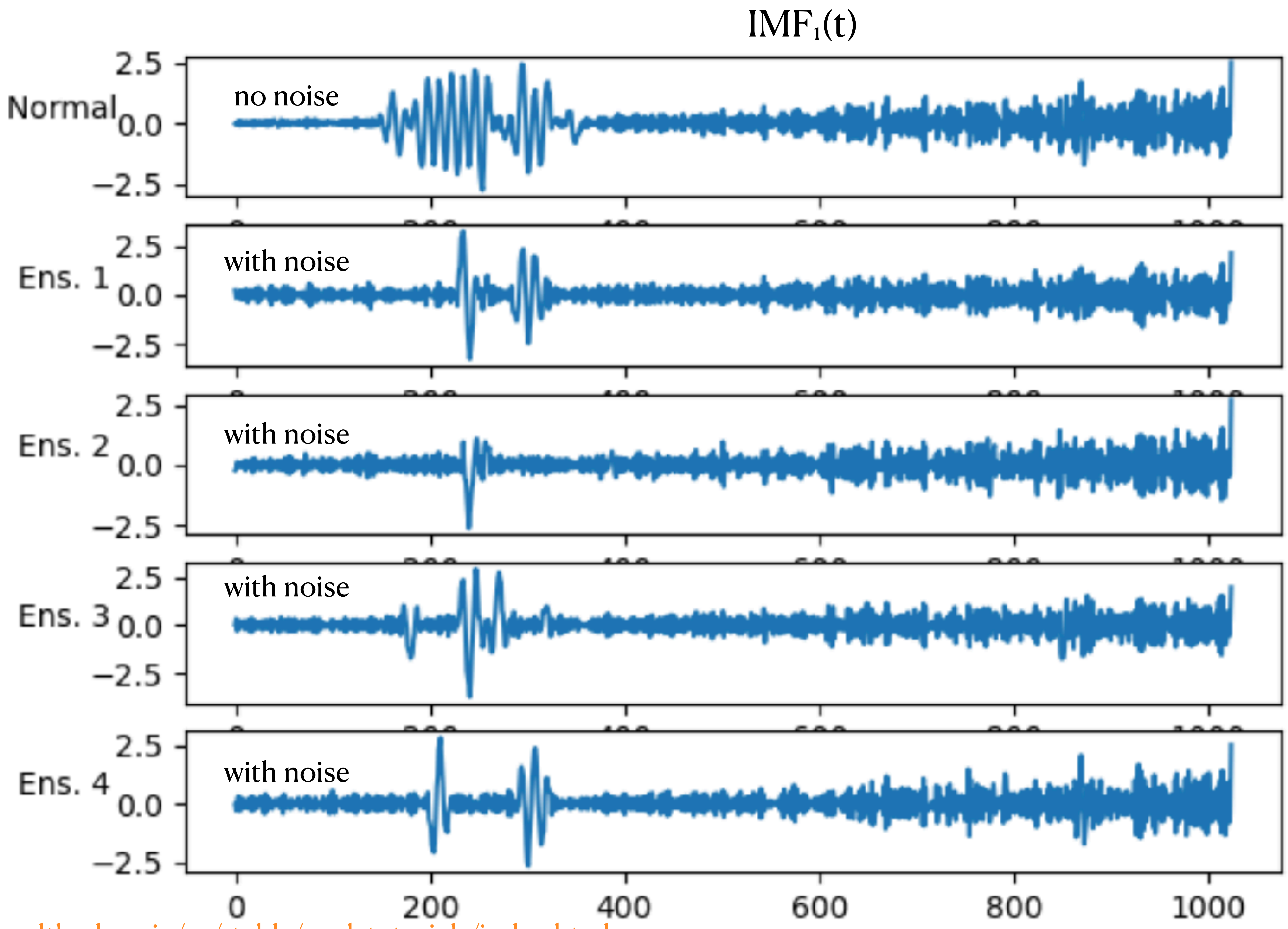
Ensemble Empirical Mode Decomposition (EEMD)

idea:

- add noise to signal L times
  to create L ensemble members

- perform EMD on
  all ensemble members

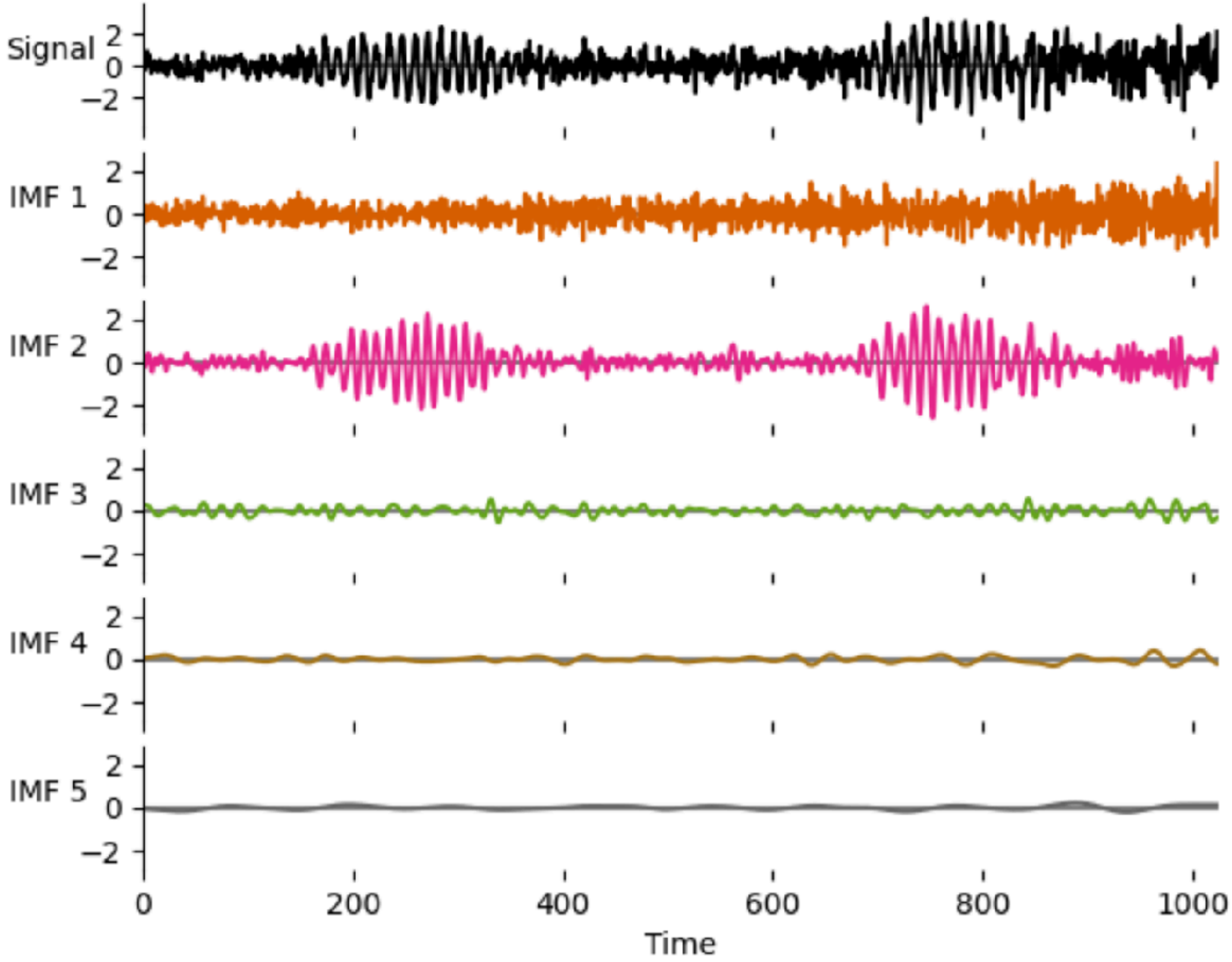
- average over all ensembles



taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

Ensemble Empirical Mode Decomposition (EEMD)

result for L=24 ensemble members



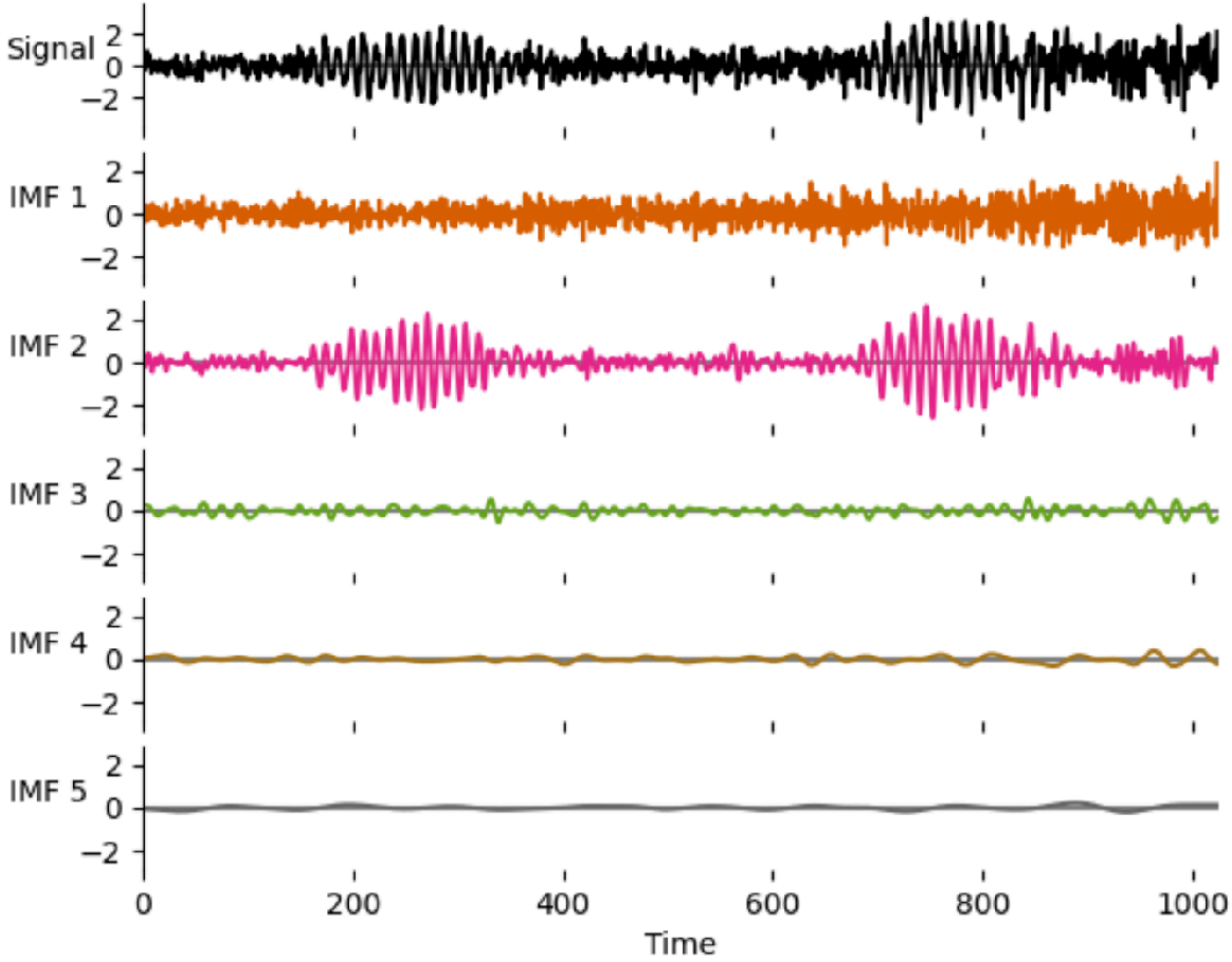
burst is separated from noise

taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

Ensemble Empirical Mode Decomposition (EEMD)

result for L=24 ensemble members

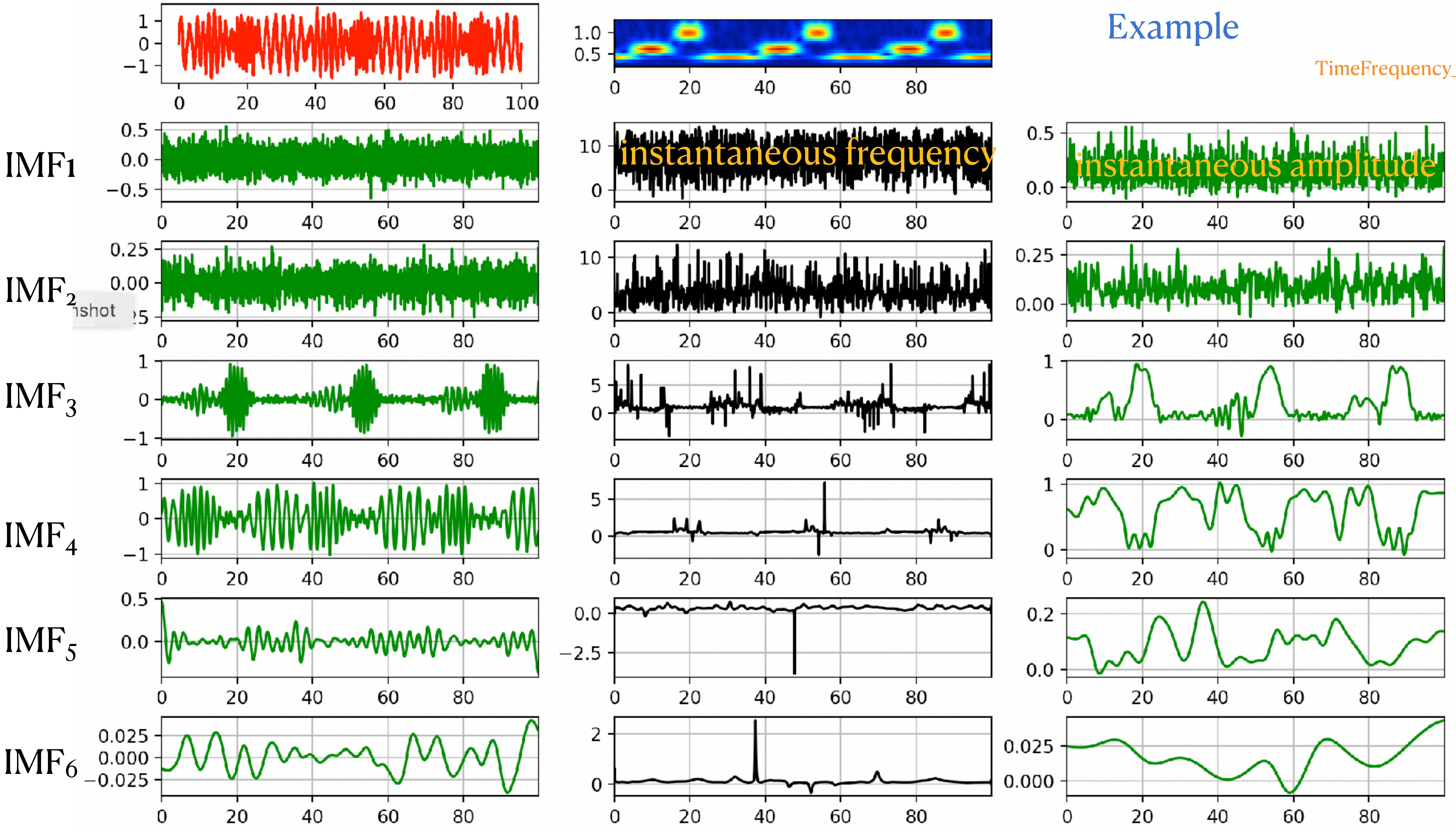
burst is separated from noise



taken from https://emd.readthedocs.io/en/stable/emd_tutorials/index.html

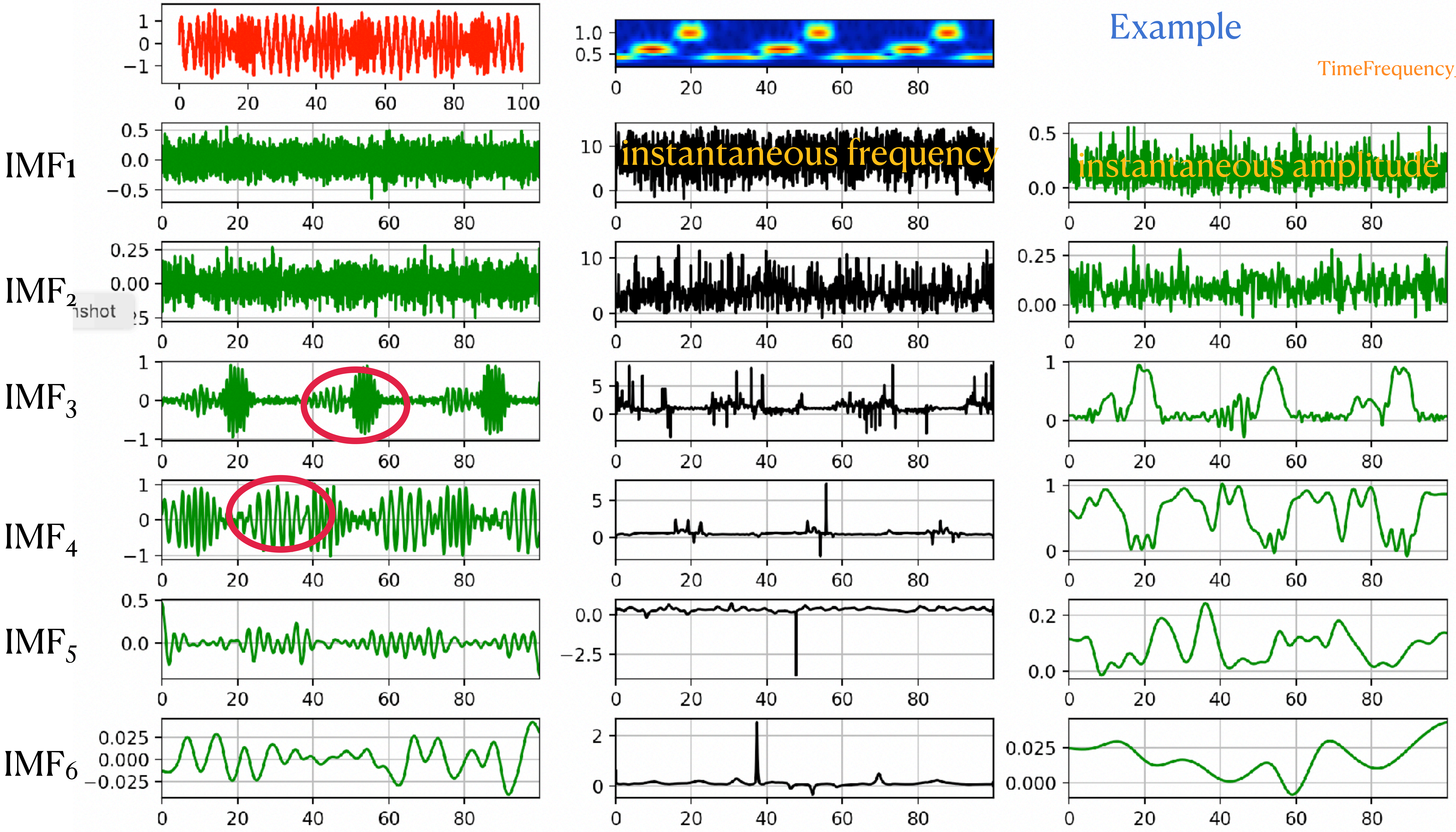
Example

TimeFrequency_5.py



Example

TimeFrequency_5.py




Thank you for your attention

good Python libraries for EMD:

EMD: https://emd.readthedocs.io/en/stable/index.html

PyEMD: https://pypi.org/project/EMD-signal/

good Python libraries for EMD:

EMD: https://emd.readthedocs.io/en/stable/index.html

PyEMD: https://pypi.org/project/EMD-signal/

any additional questions to me: write to *axel.hutt@inria.fr*