Assignment 4: IIR filters

This assignment makes use of a number of new packages:

pydicom For reading in DICOM images

scikit-image For image processing

You probably have to manually install two of these packages. To do so, open a *command terminal*, and enter the following commands:

```
pip import mat4py
pip import pydicom
```

1 Finding Filter Coefficients

- Implement the filter for the transfer function
  $y(n) = 0.7548 * x(n) - 0.7548 * x(n-1) + 0.5095 * y(n-1)$
- Apply it to an impulse input, and to a step input.
- What type of function is implemented here?

2 Compare FIR and IIR Filter

- Import ECG Data `ecg_hfn.mat`, using `scipy.io.loadmat`. The data file in *Matlab*-format contains a single vector with the name `ecg_hfn`, and has been recorded with a sample rate of 1000 Hz.
- Calculate coefficients for a window based FIR filter using `scipy.signal.firwin` to remove high frequency noise.
- Calculate coefficients for a butterworth IIR filter using `scipy.signal.butter` to remove high frequency noise.
- Adjust the filter order for both filters, and display them on the command line.
- Superpose all three signals in a plot.

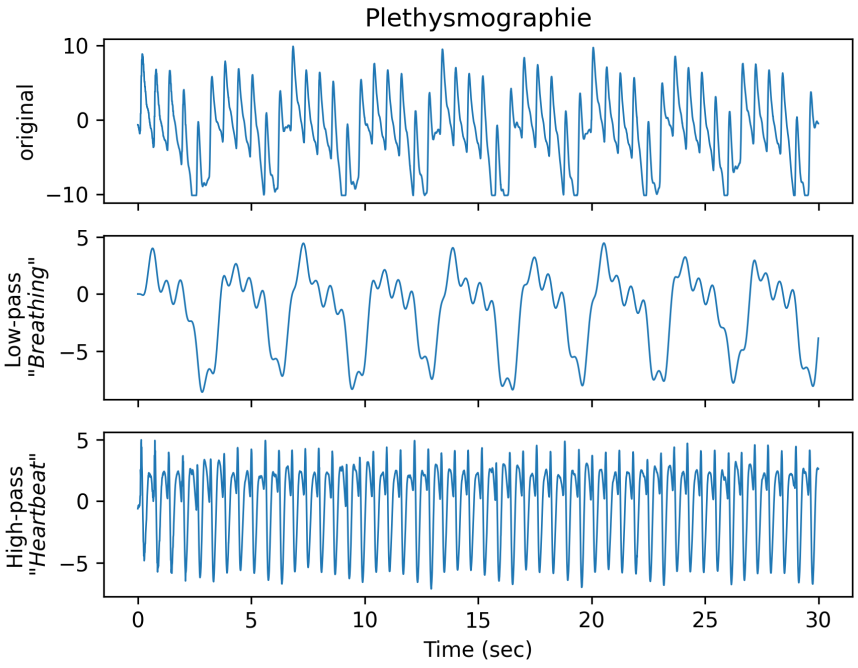
3 Pulse-Oximetry

You are a member of a team that wants to investigate the potential for a simple, non-invasive measure of respiratory effort based on the pulse oximeter signal - the photoplethysmogram or 'pleth'. (See also Addison2017.pdf.)

- The data in `pleth.pickle` contain a dictionary with a plethysmography recording from a patient monitor.
- Read in the data, using `pickle`. For this the Python-command `with` can be very helpful (you may want to look up that command in the Python documentation!):

```
# From a pickled dictionary:
in_file = 'data/pleth.pickle'
with open(in_file, 'rb') as fh:
    data = pickle.load(fh)
    rate = data['rate']
    pleth = data['pleth']
```

- Use a low-pass filter to extract the signal components related to breathing.
- Use a high-pass filter for the heartbeat components. (Note: for patient monitoring, the typical frequency range is 0.67 Hz $\sim$ 40 Hz.)
- Plot both signals in separate subplots. The result should look approximately as shown below.
- Can you also apply a bandpass-filter?



4 Image Processing - Line Detection

A good (introduction) to image processing in Python is given in Hands-on Signal Analysis with Python, pp. 96-102

- Read in the DICOM-data from `MR_MONO2-16-knee`, using `pydicom.read_file`. This gives you the image data in the property `pixel_array`.
- Display the data.
- Find edges using `skimage.filters.sobel`, and display them in a second plot (in the same figure).