names: Ricardo Leão Cardoso, Annabelle Ducret, Alexandrine Jasmine Annick Dif

The objective of this exercise is to study the influence of the parameterization of the Welch spectral estimator in order to highlight a 50 Hz perturbation in an ECG signal.

In [1]:
```
import numpy as np
import pylab as py
py.ion()
py.close('all')
import scipy.signal as sp
```

In [2]:
```
x = np.genfromtxt('ecg.dat')
fs = 500
```

Objective: Compare spectral estimation for different window lengths using welch estimation.
Plot the log spectrum of the signal using windows of 100, 500, 2000.

Q: Comment the results.

A peak is present at 50Hz on the three graphs. The first graph, with the shorter sample length (100), is the smoothest because less FTTs are used to average the signal power spectral density ((2000-50)/50 = 39), which decreases the noise. However, the spectral resolution is not precise enough to distinguish clearly the 50Hz frequency. The more the window length is increased, the more the graph becomes noisy. The second graph has more noise than the first, but the 50Hz is still distinguishable, and the spectral resolution is better. In the third graph, the spectral resolution is the highest, but there is no noise reduction.

Q: Which windows length is the most suitable for the observation of 50 Hz?

The most suitable window length is the second one (500).

Q: Why?

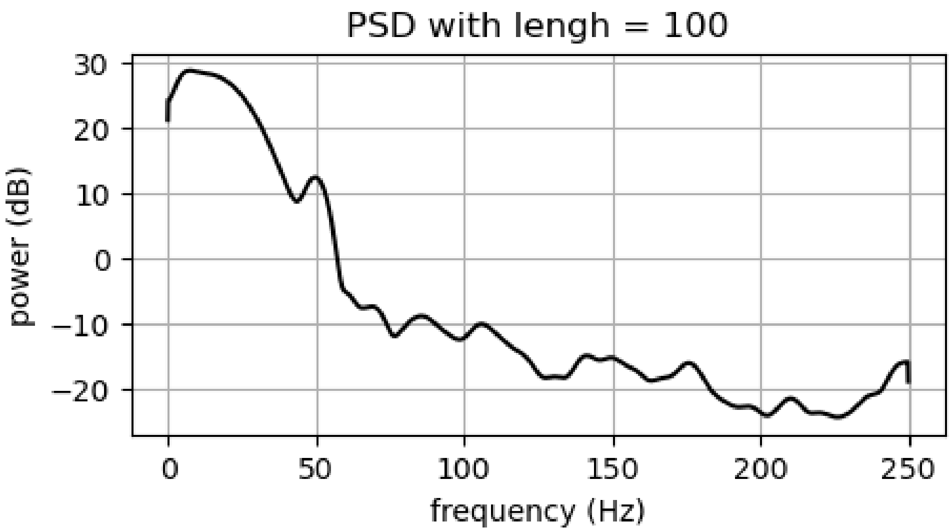
The second one provides a distinguishable 50Hz peak and a high spectral resolution.

In [3]:
```
f,X_100 = sp.welch(x, nperseg=100, nfft=4096, fs=fs)
f,X_500 = sp.welch(x, nperseg=500, nfft=4096, fs=fs)
f,X_2000 = sp.welch(x, nperseg=2000, nfft=4096, fs=fs)
```

In [4]:
```
py.figure(1, figsize=[5,8])
py.clf()
py.subplot(3,1,1)
py.plot(f, 10*np.log10(X_100), 'k')
```

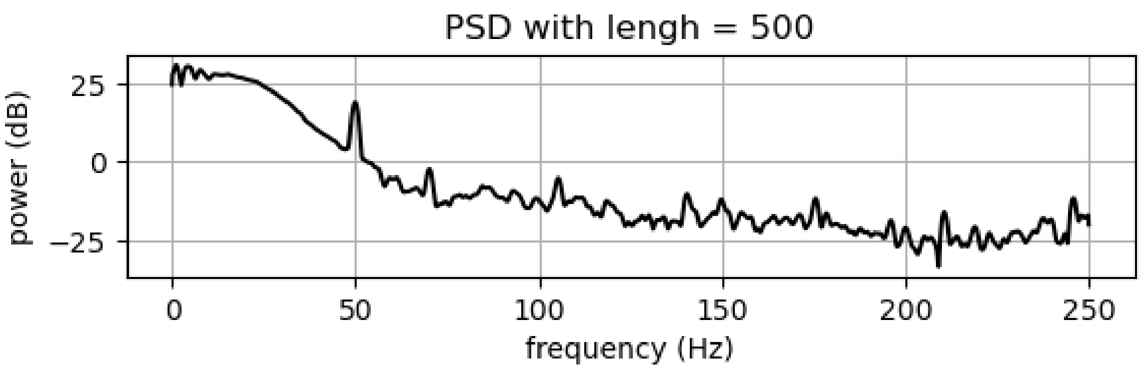
```
py.grid()
py.xlabel('frequency (Hz)')
py.ylabel('power (dB)')
py.title('PSD with lengh = 100')
```

Out[4]: Text(0.5, 1.0, 'PSD with lengh = 100')



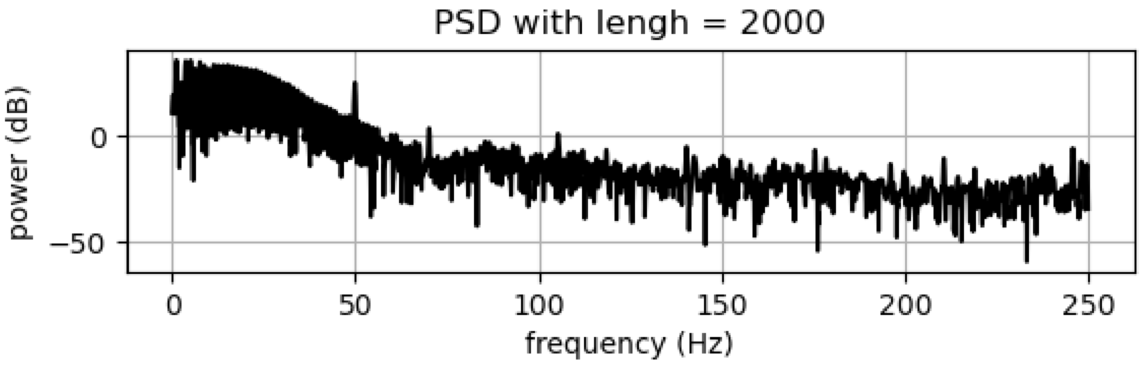
In [5]:
```
py.subplot(3,1,2)
py.plot(f, 10*np.log10(X_500), 'k')
py.grid()
py.xlabel('frequency (Hz)')
py.ylabel('power (dB)')
py.title('PSD with lengh = 500')
```

Out[5]: Text(0.5, 1.0, 'PSD with lengh = 500')



In [6]:
```
py.subplot(3,1,3)
py.plot(f, 10*np.log10(X_2000), 'k')
py.grid()
py.xlabel('frequency (Hz)')
py.ylabel('power (dB)')
py.title('PSD with lengh = 2000')
```

Out[6]: Text(0.5, 1.0, 'PSD with lengh = 2000')


PSD with lengh = 2000
power (dB)
0
−50
0
50
100
150
200
250
frequency (Hz)