Exercise 1

Oscar Björkgren 38655

Part A

i)

In this task I computed the filter order of four IIR filters in matlab. The filter order was calculated for following filters: elliptic, butterworth, chabyshev 1 and chebyshev 2. Each filter order calculation was done with the corresponding order calculation function which required passband edge frequency, stopband edge frequency, passband deviation and stopband attenuation.

Filter	Order
Elliptic	8
Butterworth	58
chebyshev 1	17
chebyshev 2	17

Table 1: Result of filter order calculations

As seen in table 1, the elliptic filter had the lowest order so it will be used as the low pass filter. The filter was constructed using the provided parameters and gave a frequency repsonse magnitude and phase visualised in figure 1.

ii)

In this task the elliptic filter is applied on the provided audio file to fitler out some noise. Using the filter parameters from the previous task I apply them to the filter function together with the noisy signal. The result is plotted in figure 2.

The result is good when comparing the noise levels in the signals. Only some small traces are left from the noise in the filtered signal. The main problem at the this point is the phase shift that can be seen when comparing to the original signal. Some phase distortion might also have occured because according to figure 1 the frequency response of the filter is non linear.

When listening to the two signals the result is satisfying and the noise has been removed.

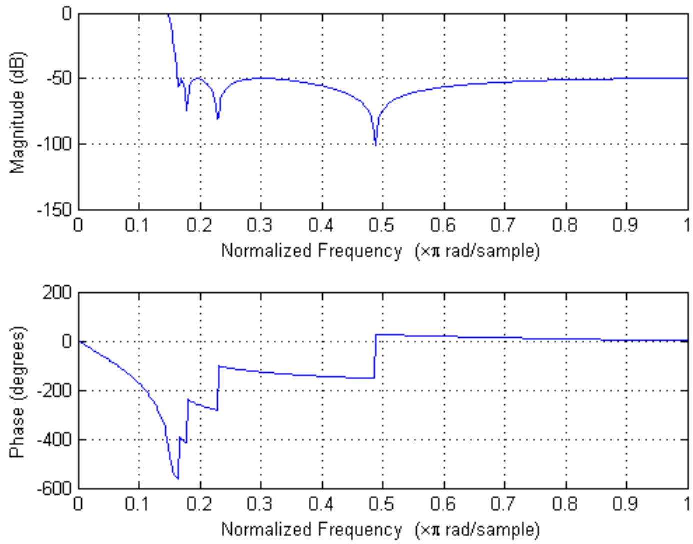

Figure 1: Frequency response

iii)

In this task the phase shift in the filtered signal is corrected to match the original signal. The function used for this is filtfilt, and the reults are plotted in figure 3. The result is quite good but the not perfect. Possibly due to phase distortion some frequencies are not corrected as well as others.

Part B

i)

In this task the signal is analyzed by looking at the fourier tranform to determine any unwanted frequency components. The fourier transform, shown in figure 4, contians 6 spikes separated by equal intervals. I interpreted the spikes as a periodic disturbance that corrupts the signal.

To find out the period I measured the distance between each spike on the x axis. The disturbing frequency components were found at 1137, 2274, 3411, 4548, 5685 and 6822.

ii)

In this task a filter is designed to remove the 6 frequency components found in the signal. This is accomplished by having a series of notch filters remove each

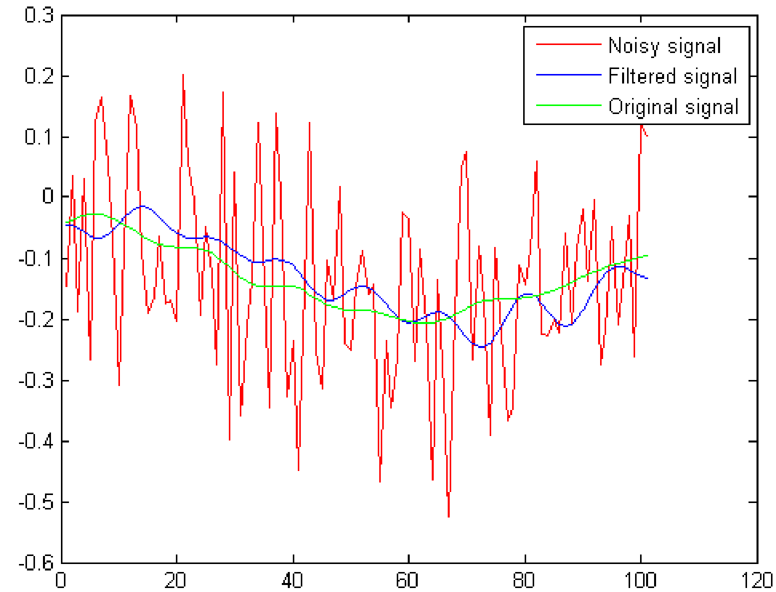

Figure 2: Plot of sequence n=1000:1100 in each signal

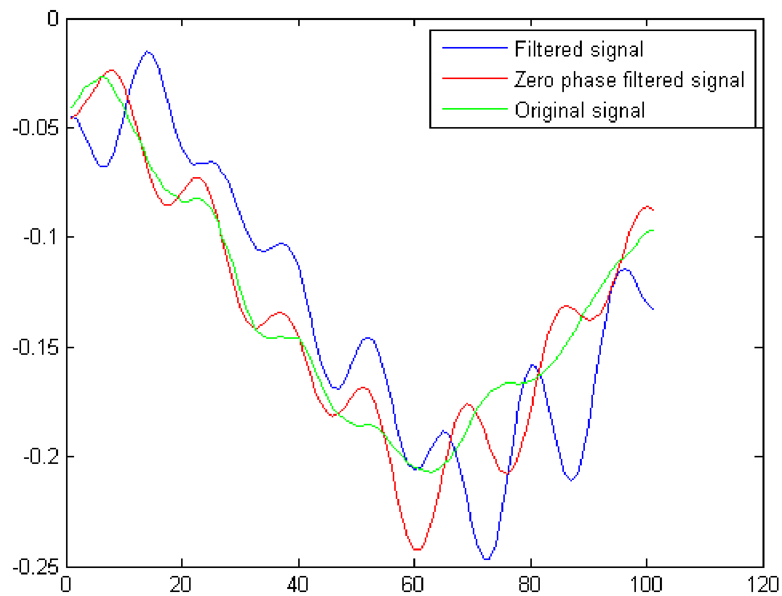

Figure 3: Plot of sequence n=1000:1100 in each signal

unwanted frequency component separately. The filter is constructed by calculating the filter coefficients and generating the filter using the coefficients and the previously filtered data. The following code is the implementation in matlab.

```
Fs = 8192;
s = load(signal.dat);
fb = 10/Fs;
r = 1-fb*pi;

period = 1137;
y = s;
for i=period:period:6*period,
    fnotch = i/length(s);
    B1 = [1 -2*cos(2*pi*fnotch) 1];
    A1 = [1 -2*r*cos(2*pi*fnotch) r^2];
    y = filter(B1,A1,y);
end
```

The disturbing frequency components were eliminated and the results are shown in figure 4. When listening to the filtered signal I noticed some delay in the filter, but otherwise it was a good result.

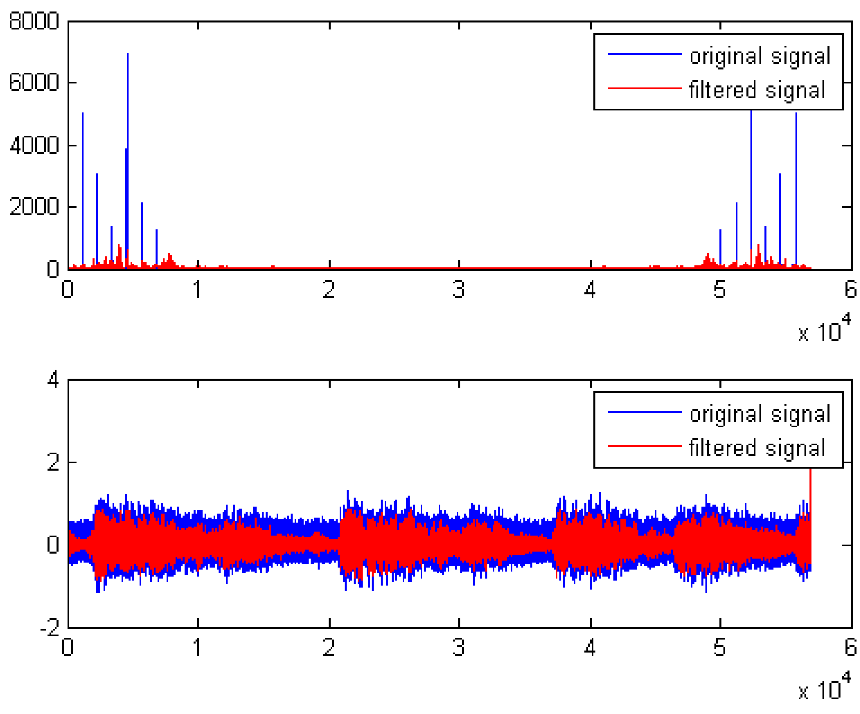

Figure 4: FFT and plot of the original and filtered signal