Aarhus Institue of Technology

Author: *Morten Hgsberg*

Date: November 30, 2021

Filtre

This project utilizes two types of filter a FIR (finite inpulse responds) filter and an IIR (infinite impulse responds) filter. The FIR filter is designed and implemented in matlab using the window method.

FIR Filter Design

First step in designing a FIR filter is to design an ideal IIR filter before trucating it with by multiplying the IIR filter with a finite length window function.

By using our spectral analysis from the earlier sections, we qualitatively decided to make the cutoff frequence $f_c = 10$. The sample frequency f_s is given from our dataset, $f_s = 47.7774$.

Lastly, the filters made order $M = 250$ thus using 250 filter coefficients.

Resolution

Next step in designing our filter, we determine the frequency resolution which provides specifications for the FIR transfer function.

$$f_{res} = \frac{f_s}{M}$$

Transfer function

Using f_{res} and f_c, we can determine which frequency bin corresponds to frequencies below f_c. This must be in done in integer values i.e. rounded to closes integer value.

$$f_{bin} = \left\lfloor \frac{f_c}{f_{res}} \right\rfloor = 52$$

In this case the bin number corresponds to f_c is 52, which we design our lowpass filter around see **??**.

These specification help determine which frequencies should be passed and which should be removed. In this case we remove everything above frequency 10.

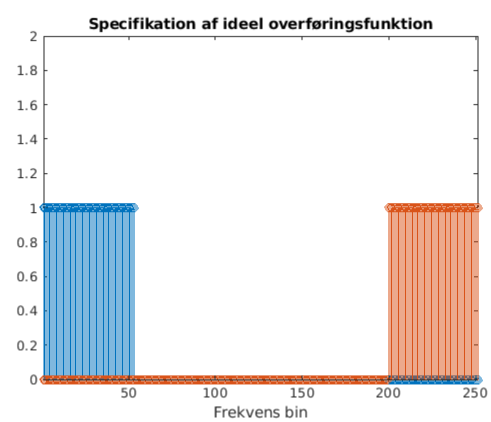

Figure 1: Specification for our transfor function

In matlab we can now make our transfer function h using the follwing code. Which when combined with, in our case a hanning window function serves to become our filter which we can run our data set through.

matlabStuff/filter.m

```
H_left  = [1 ones(1,freq_bin_round) zeros(1,(M/2)-freq_bin_round)];
H_right = fliplr(H_left(2:end));
h = fftshift(real(ifft(H)));
w_hanning = hanning(M+1)';
h_win = h.*w_hanning;

H_without_win = fft(h,f_sample_round);
H_with_win = fft(h_win,f_sample_round);
```

Figure 2: matlab code for making a transfer function

Now that we have our transfer function, we can apply it to our data to see if we can reduce the potential noice in data. we can plot its coefficients together with the window function, along with the resultat transfer function to see if it does indeed "cover" the peak around $10Hz$.

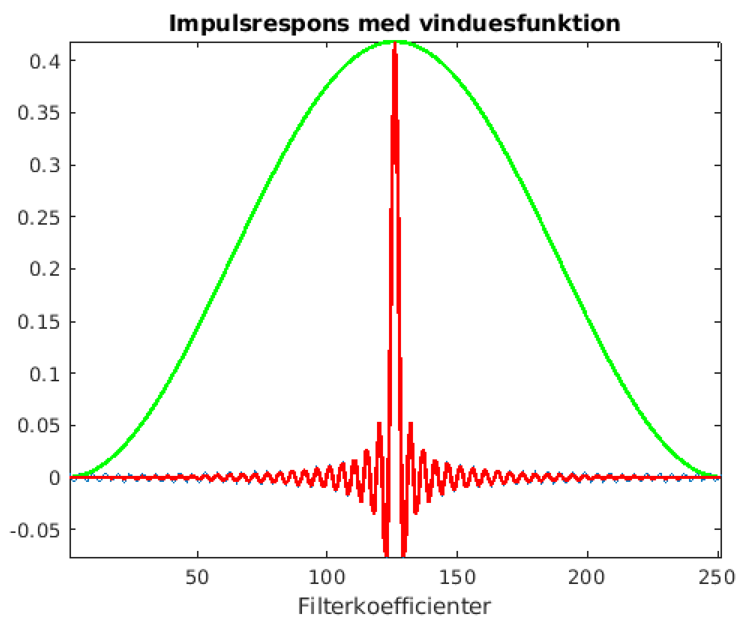

Figure 3: Impulseresponds overlayed with the window function, showing its koefficent values as functions of index numbers

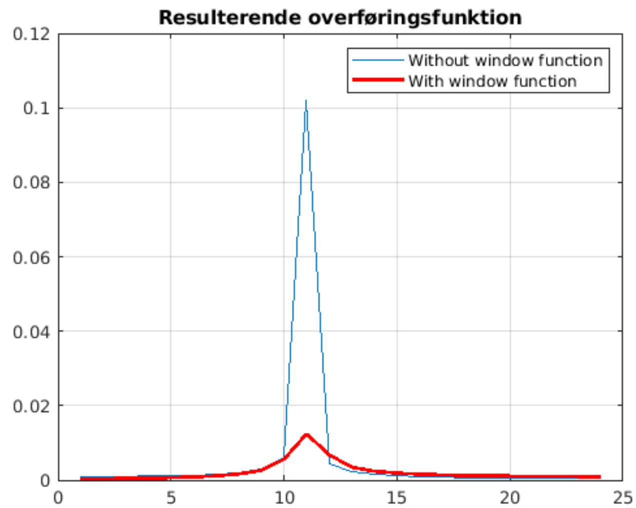

Figure 4: Resulting transfer function overlayed with and without the window function

Applying the FIR filter

With all of the preperation work done, we can now apply our filter to our dataset to see the results.

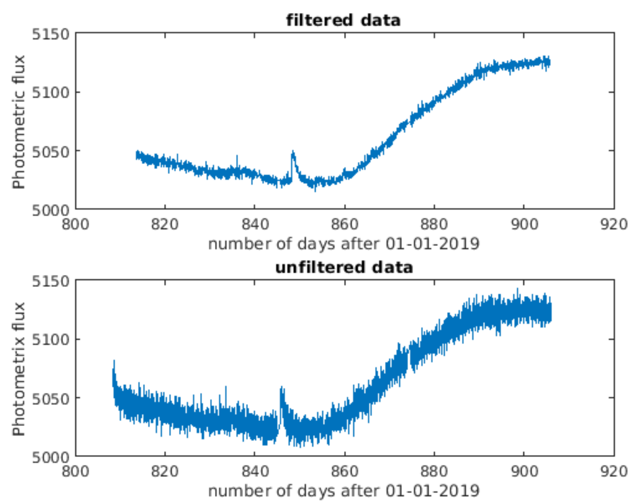

Figure 5: Side by side comparison of the data before and after the filter has been applied

Its clear fron this comparison that we have manage to remove a some of the signal noice, but since this data set does not show anything usefull, we wont be able to tell anything from this filtering.