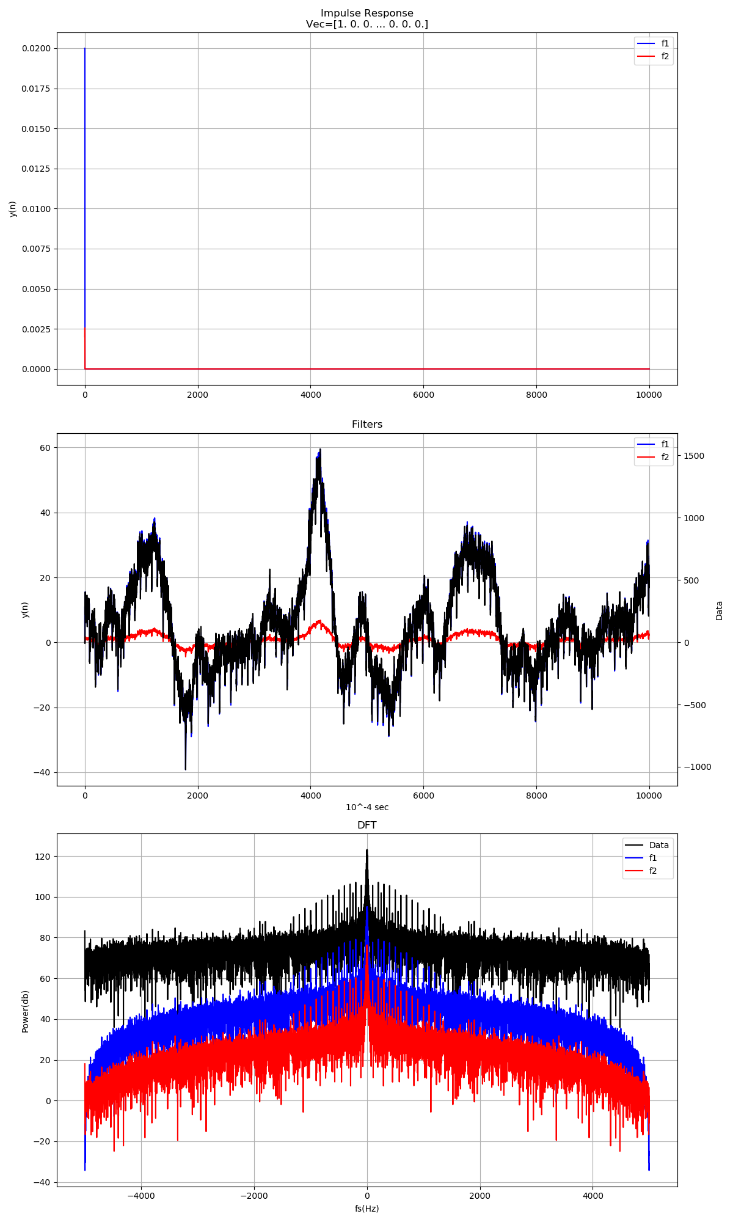
**Yuval Samoilov Katz Id 204025258**

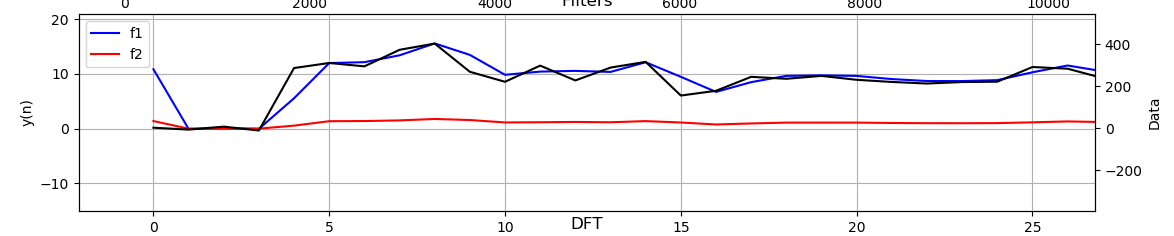
**Assignment 10 😊**

1. 4 - the filters behave like a FIR filter since it does not depend on y(n) rather on a set of finite values (i.e. k – 50 coefficients). In addition, they are both Low Pass Filters (LPF) since low frequencies are being passed while high frequencies are silent. Filter 1 (Blue) is more sensitive to fluctuations. Filter 2 (Red) is a much stronger LPF since it decreases more significantly all the signal’s frequencies. Both still leave some white noise at all frequencies.

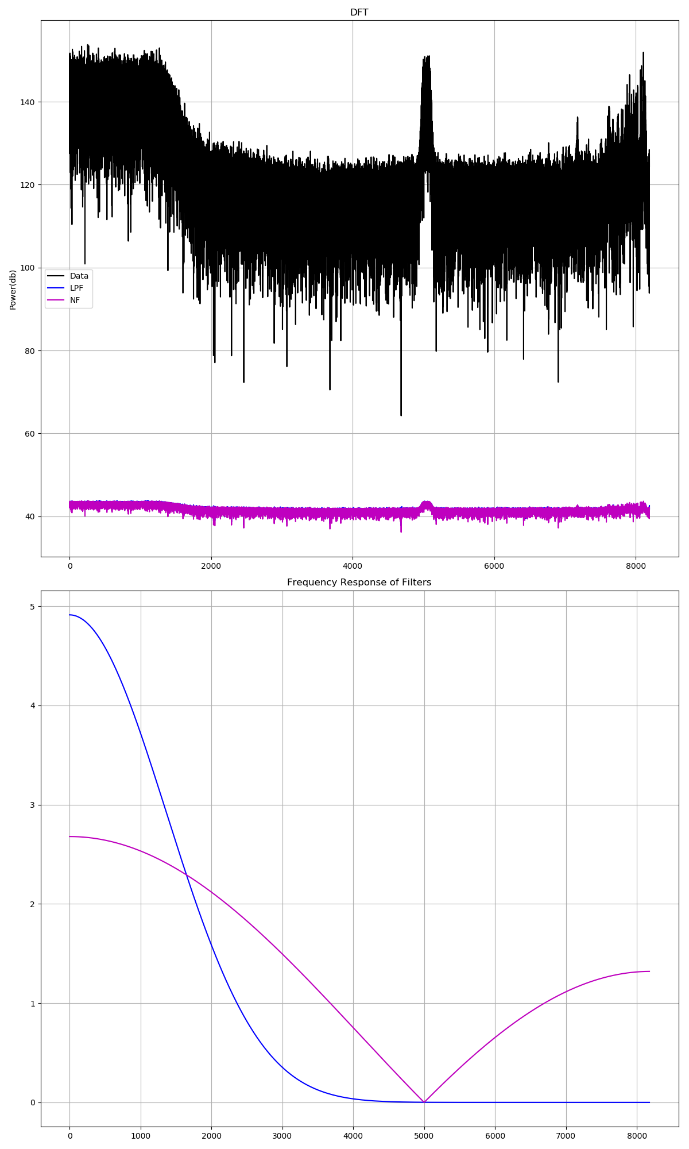
5 – to some extent, they both represent a linear combination and smooth the data.

Filter 1 (blue) fits the data (black) better at presenting trends, critical frequencies and higher resolution (in the sec fig. the overlap each other), but the edges are being poorly presented in compare to Filter 2 (Red, see zoom in on Fig.2 below). Since rectangle filter over the interesting frequencies would give a sinc function (with infinite values), an optimal filter depends on what we care about – it will have two main properties, both (a) minimum distortion, and (b) relative low filtering around critical frequencies. This can be achieved for example, by combining these two with matching weights for the ranges where each is better. i.e. Filter 2 around edges (since it achives (b) and causes a distortion of the original amplitude and relative strong decreasing, such as Chevichev filters) and Filter 1 elsewhere (since it achieves (a) e.g. similar to Butterworth filters).

**Fig. 1-2**

****

See the ex1.py code for implementation.

1. Almost 8 seconds of Sting’s song (English man in NY) were mixed with noises and sampled at ~16KHz

(b) – The signal definitely includes noises – (1) a narrow noise around at 5 KHz (symmetric so the same goes for these negative frequencies as well), and (2) additional white noise for all frequencies, and changes in power freqs below 1.3KHz and above 7.5KHz. Between them, the Amplitude in db is stable except for the narrow noise area.

(c) – the filter (Blue) is a Low Pass Filter by Butterworth 15 pole filter with fc=1.3 KHz. I stopped frequencies above 1,300 Hz, allowing only the lower ones.

See zoom in on fig 2-2 for minor differences between filters.

(d) – the filter (Purple) is a Notch Filter with fc=5 KHz. I filtered out requencies of 5,000 Hz, and did a two way LPF for both ends, stronger for higher frequencies, allowing only the lower ones

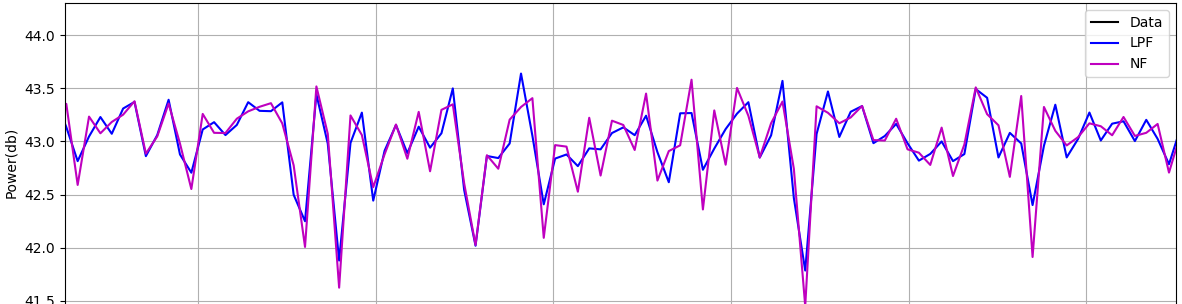
They both filtered out all the high frequencies derived from white noise among the signal. In addition, they both filtered the narrow noise around 5KHz.

I’ve saved the filtered sound and clearly a major difference is heard compared to the original signal:

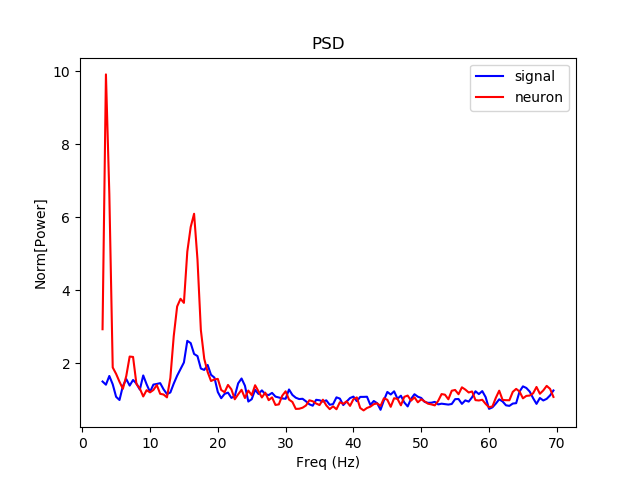
**Before** **After**

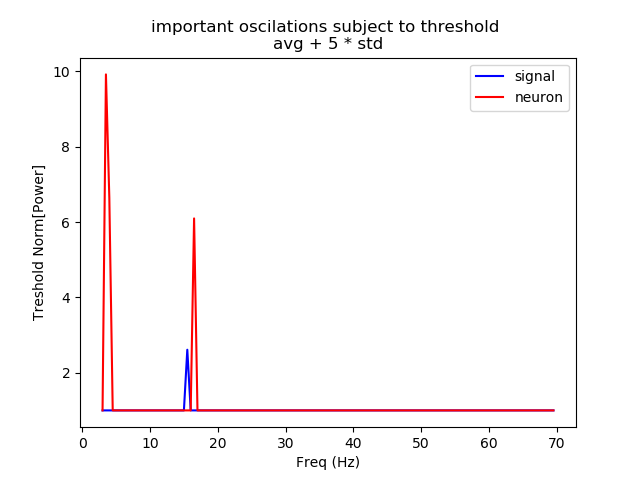
 

**Fig 2-2**

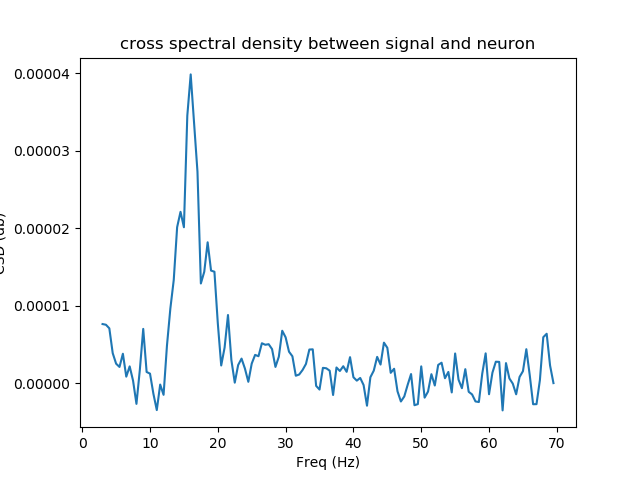


See the ex2.py code for implementation.

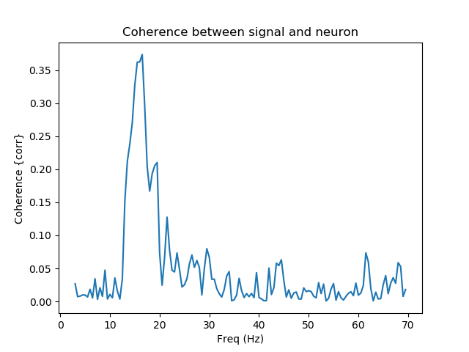
1. (a) – A signal and a neuron were recorded at 24038 samples per second for 58 seconds in total. Normalizing the signal’s energy for frequencies (avg after important signals, at 30-70Hz) is shown here by PSD.

For the range between 3-70 Hz, the Power Spectral Density (PSD) of the signal (blue) looks like white noise which accompanied the neuronal activity (red). After normalizing both the signal and the neuron, the neuron seems to have three patterns in relatively low frequencies – high frequency around 3 Hz, small one at 8 Hz and another around 18 Hz.

(b) Testing whether one of them is unique and independent of the other, so that important oscillation frequencies really exists using a rough threshold (5 std distance from average) shows that only the two higher ones are really the significant one (under this threshold) and can be characterized by two important oscillations regardless of the noise (again, subject to this threshold). Note that the noise also seems to have a significant oscillation around 15Hz, suggesting a higher noise derived from 15 Hz periodic function.



(c) – The cross spectral density between them shows a higher value for the relationship between these two signals, around 15-20 Hz. But this is sensitive to the power of the signal and the neuronal activity. This is true under the conditions we used, such as window size, overlapping and sampling frequencies.

(d) - in addition, using coherence to again distinguish the relationship between them regardless of the values, supports the existence of oscillation around 15-20Hz. A significant threshold is required to determine whether the noise affects the real signal, under the limitation we mentioned above.

See the ex3.py code for implementation.