Analysis of ECG Signals

Project 1A Alternative II

SSY130

Hanna Åvall, Isabella Sanchez, José Outomuro, Edgar Sánchez

November 2022

1 Heart Rate Estimation using the Fourier Transform

2.2.2

The visual heart rate was calculated using Equation 1, where HR is the heart rate, n_p is the number of peaks and t_i is the time interval

$$HR = n_p \times \frac{60}{t_i} \tag{1}$$

Athlete	Heart rate (BPM)
1	60
2	60
3	54
4	48
5	48
6	54
7	72
8	72
9	54
10	54
11	66
12	54
13	36
14	66
15	54
16	54
17	72
18	60
19	60
20	72
21	48
22	66
23	48
24	42
25	54
26	66
27	48
28	48

Table 1: The visual estimation of the heart rate from the lead type AVR

2.2.5

All 28 athletes heart rate was plotted along side the visual estimation. The visual estimation is shown as circles and estimation by the fbmp function as plus signs, the median of the fbmp estimation is shown as a square and the mean value of the fbmp as a diamond.

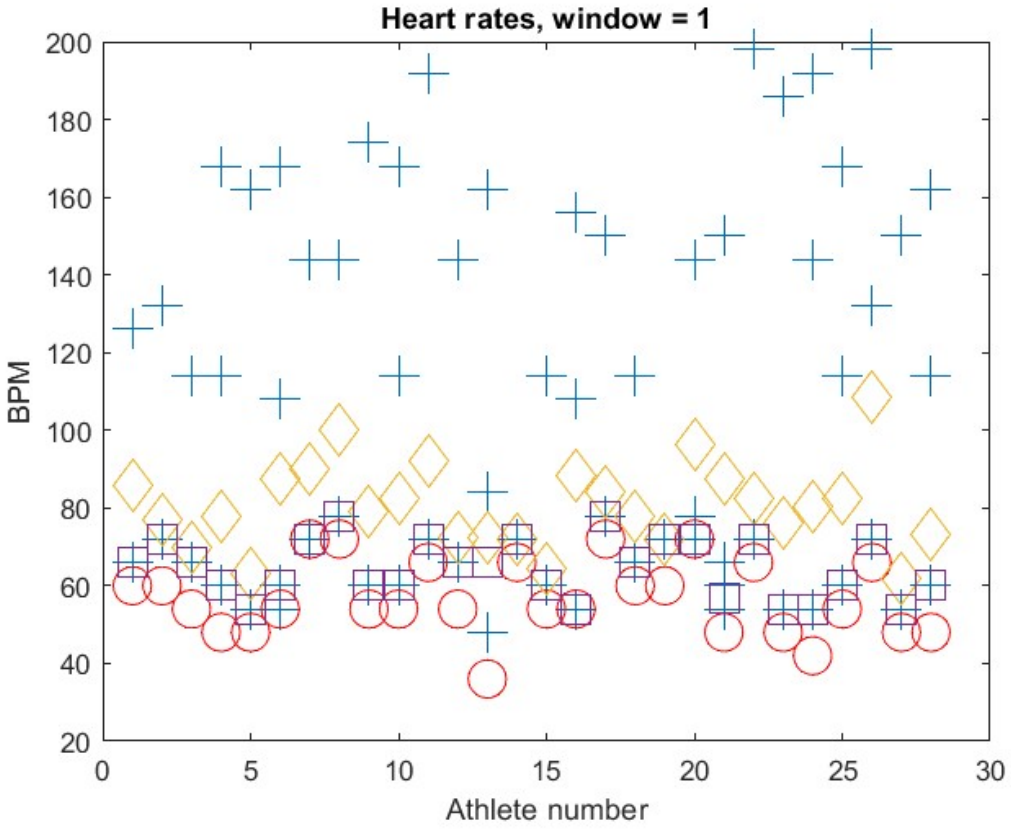

Figure 1: Plot over all 28 athletes heart rate with visual estimation and estimation by the fbmp function

In the plot it can be seen that the estimated values are wide spread for each athlete, this could be due to the leads being on different parts of the body, some are probably on places which are hard to get a signal from and therefore, we get multiple heart rates for each person.

Since the diamonds are representing the mean value in the figures and we can analyse the stacking of heart rates. Since we usually only see about 2 plus signs for each athlete, and the diamond for the first athlete for example is not in the middle of these two, but rather is one third of the distance up from the first value, which means that there are more of the exact same estimation stacked in the lower one than in the higher, shifting the mean value down.

2.2.6

The plot in Figure 1 is recreated with the different window functions, Gaussian in Figure 2, Taylor in Figure 3 and Chebyshev in Figure 4.

We can see that different windows produce varying results. Gaussian and Chebyshev are very similar. The Gaussian contains one more value at low heart rates, Chebyshev one more at high heart rates. Chebyshev captures more values than the rectangular window. This probably has something to do with how the lobes look in the frequency domain.

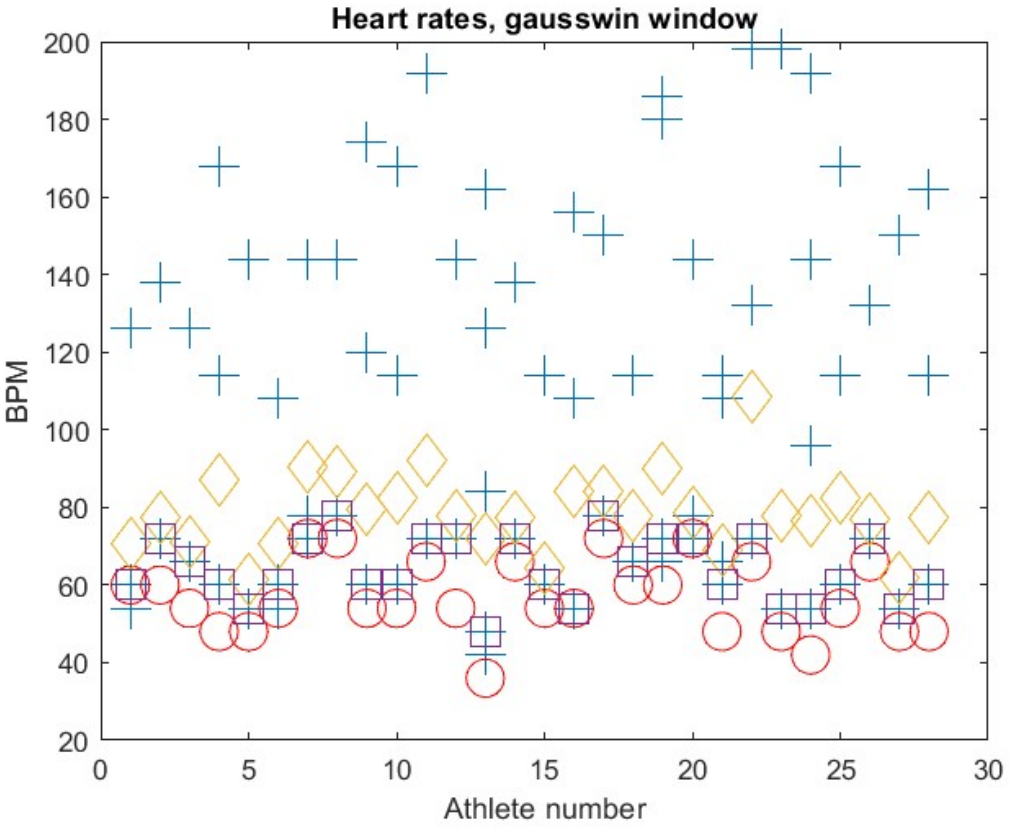

Figure 2: Plotresults after using a Guassian window in fbpm function

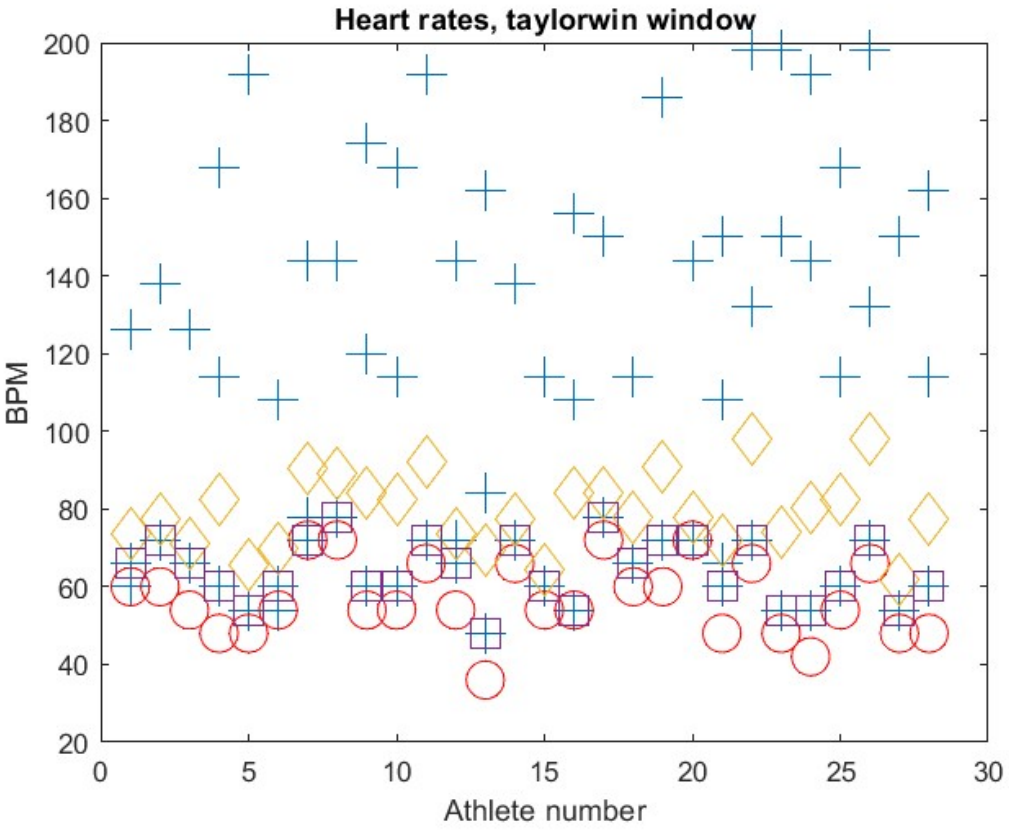

Figure 3: Plotresults after using a Taylor window in fbpm function

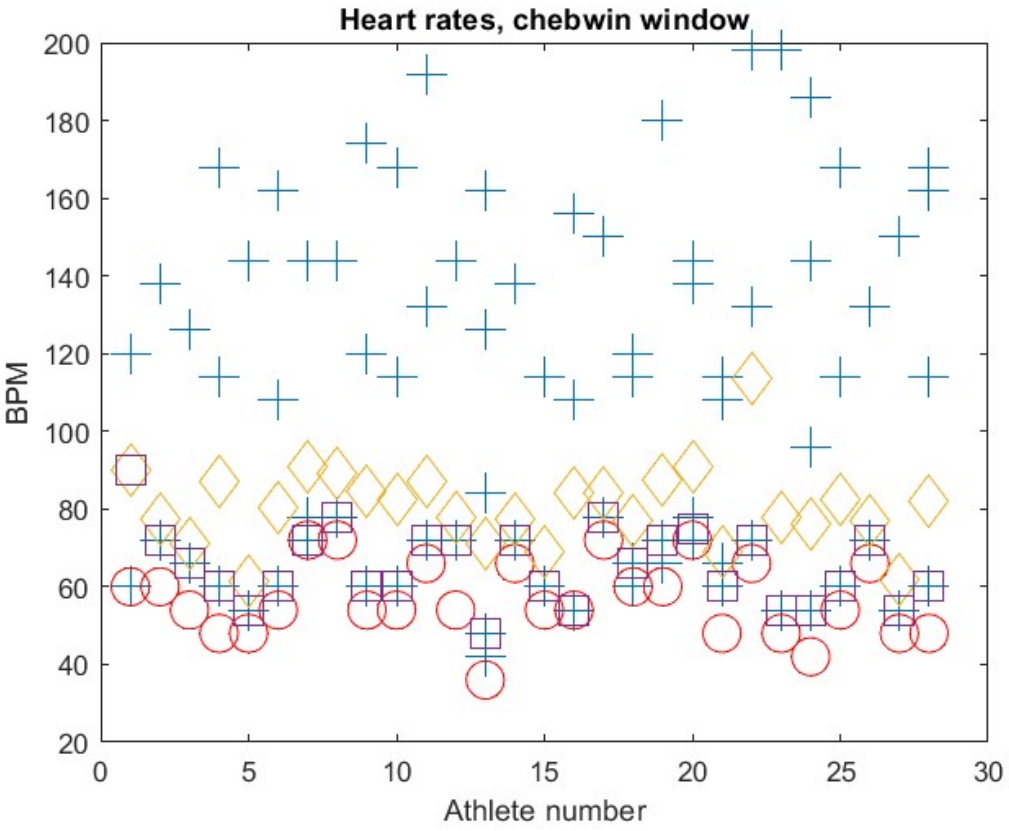

Figure 4: Plotresults after using a Chebychev window in fbpm function

2.3.1

N_h shows the number of harmonics. The sample rate, f_s, decides how many samples N we get. $N_h = N - 1$. A high sample rate gives narrow harmonic lobes and vice versa. We look at the heart rate, HR, which is given by the lowest frequency and is represented by the basic harmonic. Amplitude will be affected by number of harmonics as well.

2.3.2

$$x(1) = x(0) = X(W_0) \tag{2}$$

$$X(Ndtft) = X(Ndtft - 1) = X((Ndtft - 1) * 2\pi f_s/Ndtft) = X(W_{Ndtft-1}) \tag{3}$$

2.3.3

Show that the DTFT of the signal in (3) has the form.

$$X(\omega) = \beta_0\delta(\omega) + \sum_{k=1}^{N_h} \beta_k\delta(\omega - k\omega_0) + \beta_k^*\delta(\omega + k\omega_0)$$

We have our original discrete function in (3) as follows.

$$x(n) = \sum_{k=0}^{N_h} a_k cos(k\omega_0\Delta tn + \phi_k)$$

as we know from theory, the cosine can also be defined as

$$cos(x) = \frac{e^{jx} + e^{-jx}}{2}$$

Hence, we can view $X(\omega)$ as follows:

$$DTFT[x(n)] = \sum_{-\infty}^{\infty} x(n)e^{-j\omega\Delta tn} = DTFT[x(n)] = \sum_{k=0}^{N_h}\sum_{-\infty}^{\infty} a_k cos(k\omega_0\Delta tn + \phi_k)e^{-j\omega\Delta tn} =$$

$$\sum_{k=0}^{N_h} \frac{2a_k\pi}{2} \sum_{-\infty}^{\infty} (e^{j(k\omega_0\Delta tn+\phi_k)} + e^{-j(k\omega_0\Delta tn+\phi_k)})e^{-j\omega\Delta tn}$$

$$\sum_{k=0}^{N_h} \pi a_k \sum_{-\infty}^{\infty} e^{j(k\omega_0\Delta tn)}e^{-j\omega\Delta tn}e^{j\phi_k} + e^{-j(k\omega_0\Delta tn)}e^{-j\omega\Delta tn}e^{-j\phi_k}$$

From inspection it is evident that $e^{-j\phi_k}$ and $e^{j\phi_k}$ are independent of n and thereupon can be taken out of the transform equation, leaving us with:

$$\sum_{k=0}^{N_h} (\pi a_k e^{j\phi_k} \sum_{-\infty}^{\infty} e^{j(k\omega_0\Delta tn)}e^{-j\omega\Delta tn} + \pi a_k e^{-j\phi_k} \sum_{-\infty}^{\infty} e^{-j(k\omega_0\Delta tn)}e^{-j\omega\Delta tn})$$

From the Transform Pairs table located in the lecture 2 notes, we know that:

$$DTFT[e^{j\omega_0 \Delta t n} x(n)] = X(\omega - \omega_0)$$

By making the corresponding substitution inside the transform equation, we get that:

$$\sum_{k=0}^{N_h} \pi a_k e^{j\phi_k} \delta(\omega - k\omega_0) + \pi a_k e^{-j\phi_k} \delta(\omega + k\omega_0)$$

When looking at the a_k values that affect each delta function, we notice that the phase on the right is just the conjugate of the one on the left, relating as such:

$$(\pi a_k e^{j\phi_k})^* = \pi a_k e^{-j\phi_k}$$

We can name this as a single variable for the sum, β for example, making our final transform look like:

$$\sum_{k=0}^{N_h} \beta_k \delta(\omega - k\omega_0) + \beta_k^* \delta(\omega + k\omega_0)$$

If we evaluate the first term of the sum where k=0, we reach the next equation:

$$\beta_0 \delta(\omega) + \sum_{k=1}^{N_h} \beta_k \delta(\omega - k\omega_0) + \beta_k^* \delta(\omega + k\omega_0)$$

Which is equal to the representation we had been looking for, hence Q.E.D.

2 Harmonic disturbance cancellation using Local Models FIR design

3.1.1

Adding disturbance

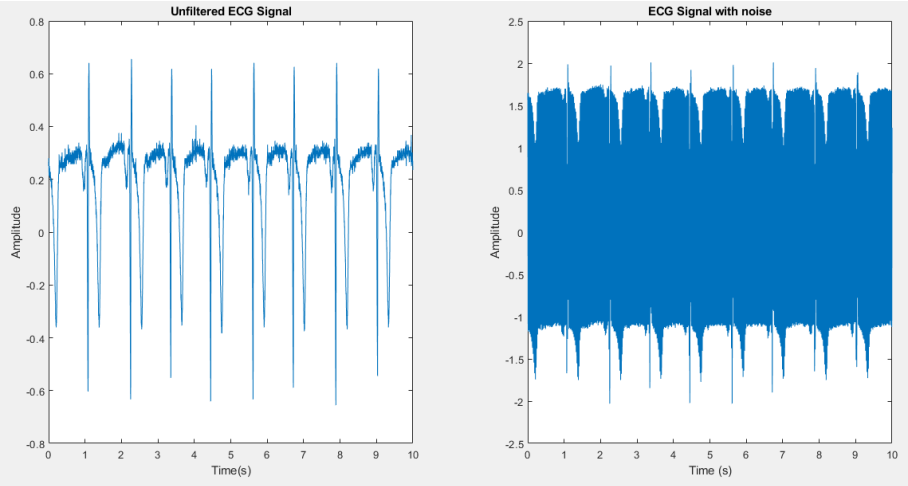

Figure 5: Unfiltered ECG Signal (left) vs. ECG Signal with disturbance with gain=0 added (right).

As expected, once the disturbance is added it is harder to be able to distinguish the information along the input signal. That is why the filter to remove noise must be apply, to remove the added noise and interpret clinical information from the input signal later on.

3.1.3

Results adding gain=0 disturbance

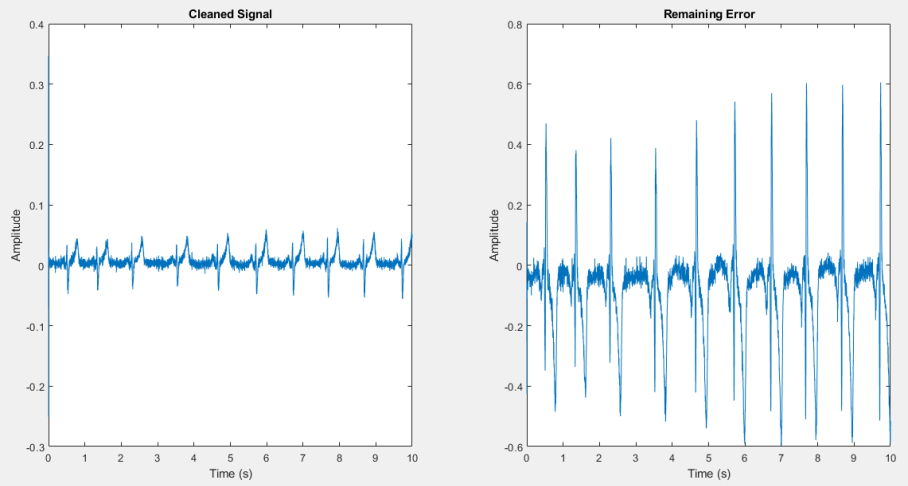

Figure 6: Cleaned Signal with Gain=0 and M=1 (left) vs. Remaining Error (right).

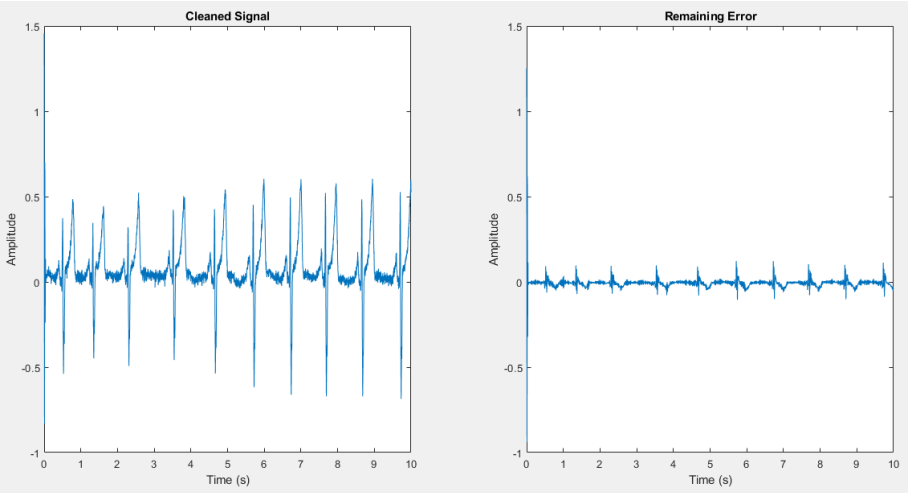

Figure 7: Cleaned Signal with Gain=0 and M=10 (left) vs. Remaining Error (right).

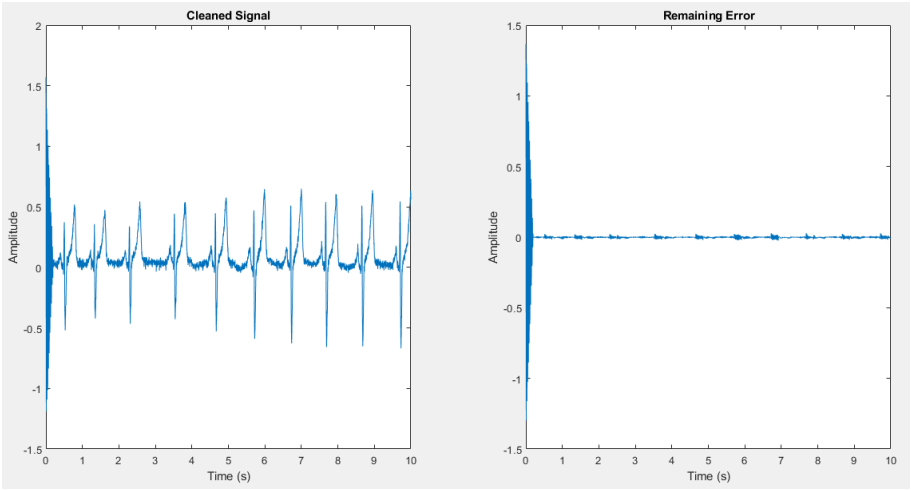

Figure 8: Cleaned Signal with Gain=0 and M=50 (left) vs. Remaining Error (right).

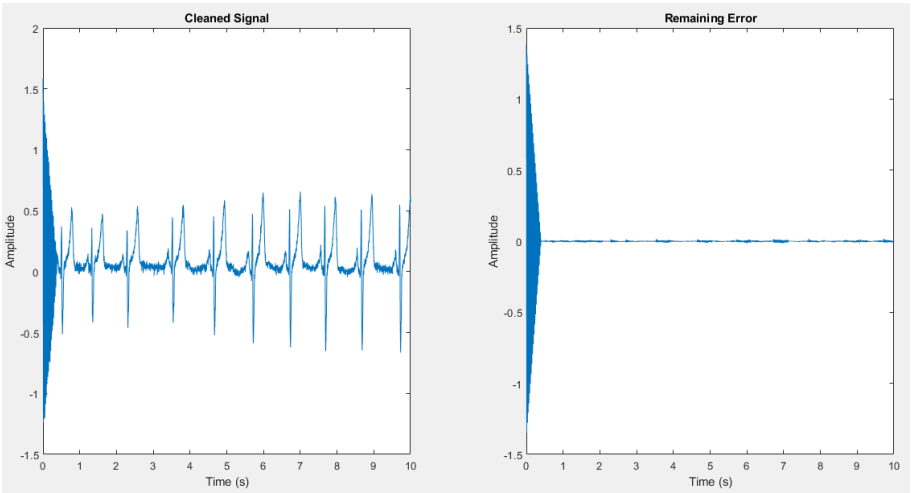

Figure 9: Cleaned Signal with Gain=0 and M=100 (left) vs. Remaining Error (right).

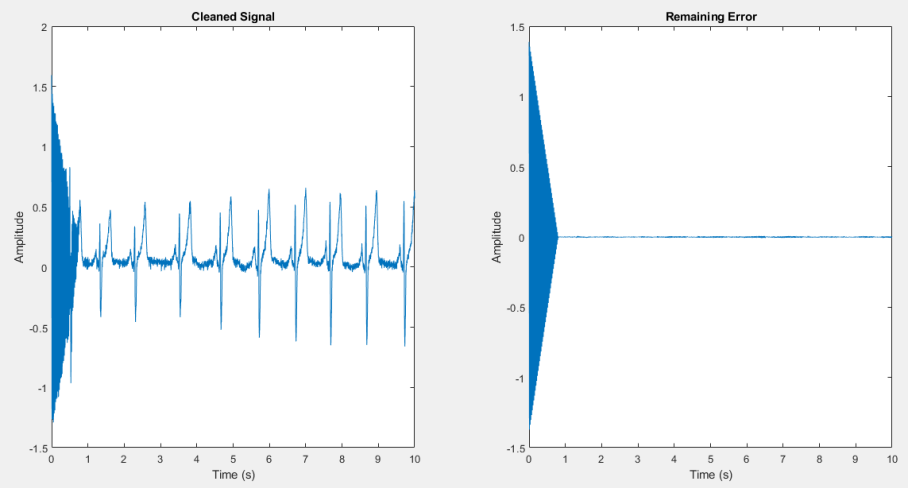

Figure 10: Cleaned Signal with Gain=0 and M=200 (left) vs. Remaining Error (right).

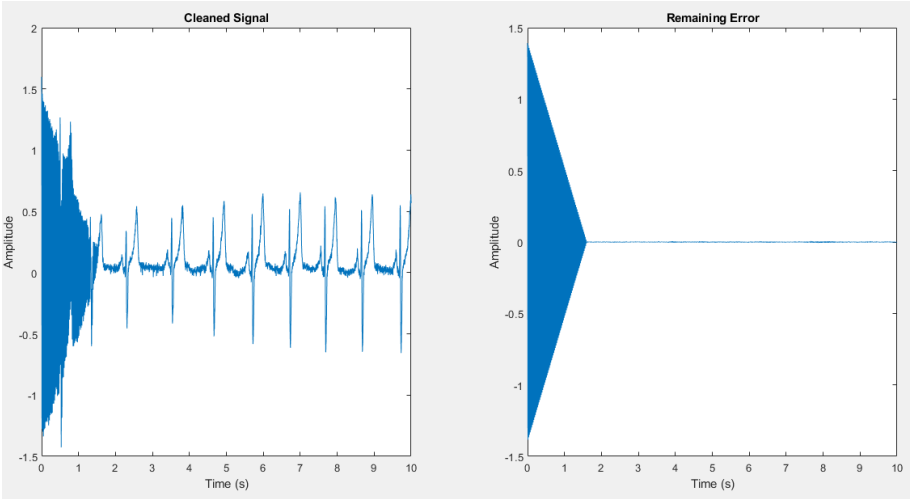

Figure 11: Cleaned Signal with Gain=0 and M=400 (left) vs. Remaining Error (right).

M	1	10	50	100	200	400
RMS	0.179	0.0209	0.0059	0.0038	0.0024	0.0013

Table 2: M and RMS values with gain=0

Considering the obtained root mean square (RMS) values, as can be seen on Table 2, the bigger the M value is, the smaller the RMS is. That is happening because the noise with gain=0 is fixed over time. Thus, increasing M a smaller remaining error is obtained. At the same time, when we increase M, the main lobe will get wider, in other words, the model disturbance will increase. By looking at the figures, M = 10 seems to be a good trade of between error and disturbance.
Another interesting observation that could be made was that increasing the M value also the disturbance around 0 in the remaining error plot increased. Thus, to determine the adequate value of M it should be take into account what is the desired result of it. Then choose if it is better to have a wider main lobe and lower remaining error or the other way around.

3.1.4

Results adding gain=0.001 disturbance

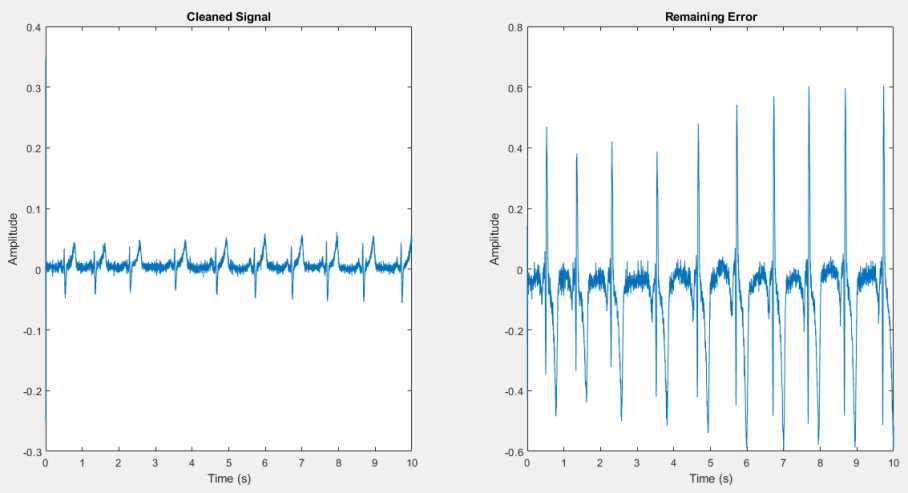

Figure 12: Cleaned Signal with Gain=0.001 and M=1 (left) vs. Remaining Error (right).

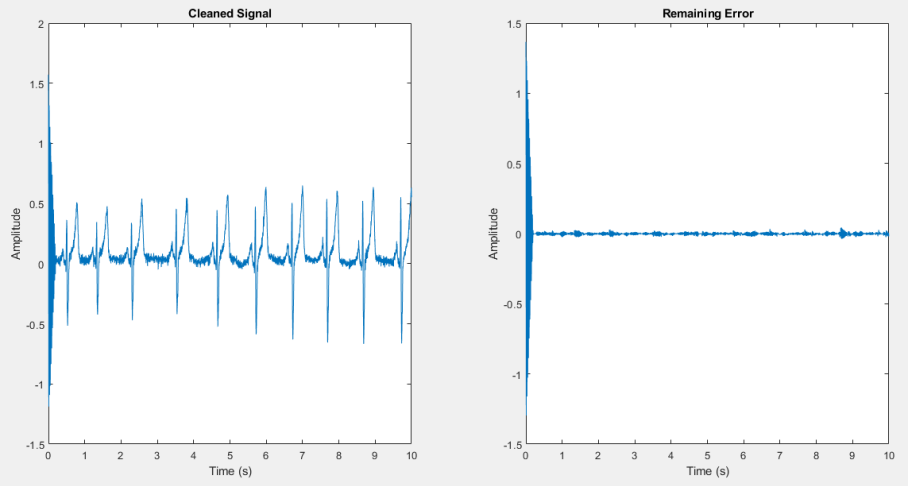

Figure 13: Cleaned Signal with Gain=0.001 and M=50 (left) vs. Remaining Error (right).

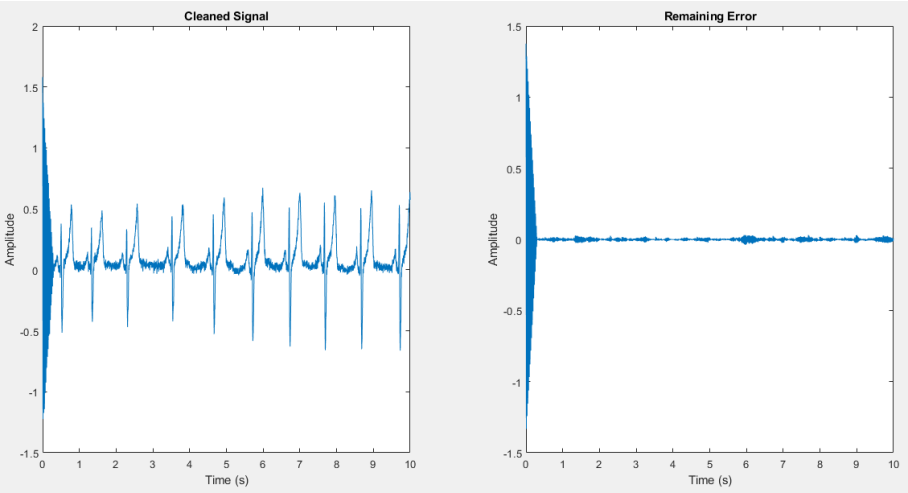

Figure 14: Cleaned Signal with Gain=0.001 and M=75 (left) vs. Remaining Error (right).

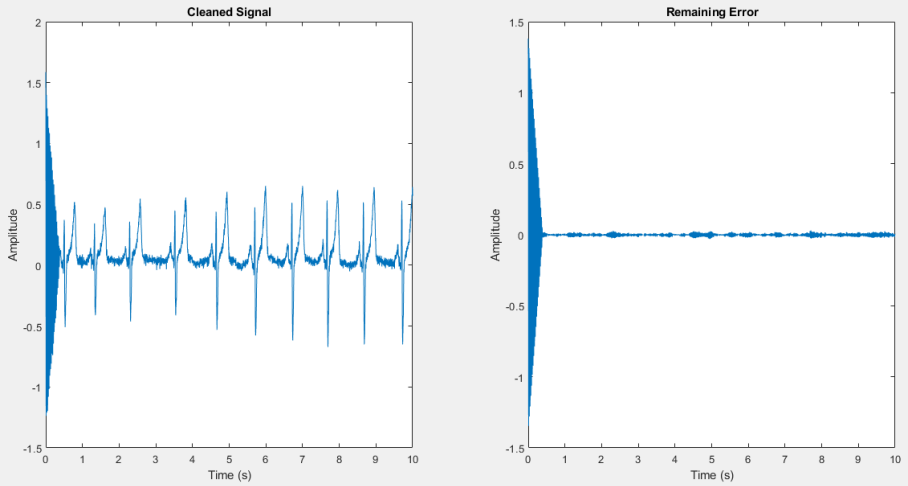

Figure 15: Cleaned Signal with Gain=0.001 and M=100 (left) vs. Remaining Error (right).

M	1	50	75	100
RMS	0.179	0.0085	0.0084	0.0088

Table 3: M and RMS values with gain=0.001

Once the gain has been added to the disturbance, a new behaviour can be observed. In this case, with gain=0.001, a bigger M value does not get a smaller RMS value for all cases. The reason behind that fact is that, as the disturbance is not fixed in time anymore, even though a bigger M value attenuates better the noise, the model error gets bigger as well. Therefore, it is needed to look for the convenient value of M. As can be seen on Table 3, with gain=0.001, the convenient M value was set around 75.

3.1.5

Results adding gain=0.001 disturbance

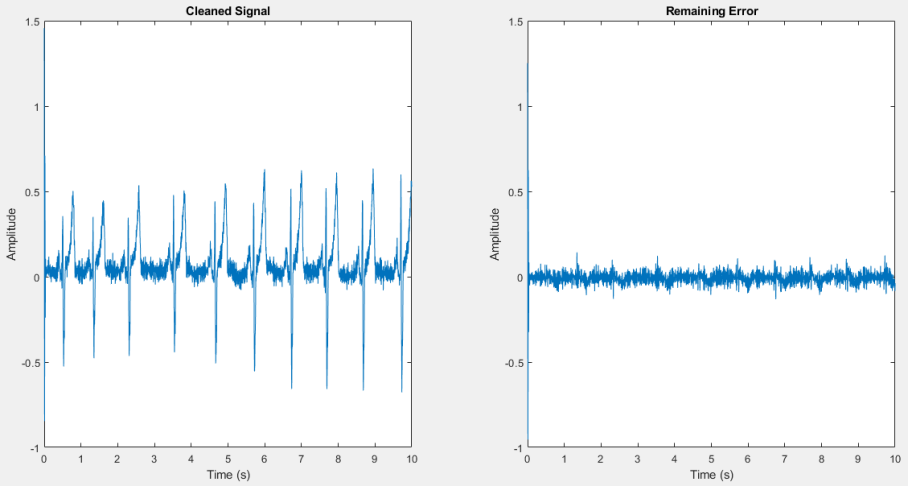

Figure 16: Cleaned Signal with Gain=0.01 and M=10 (left) vs. Remaining Error (right).

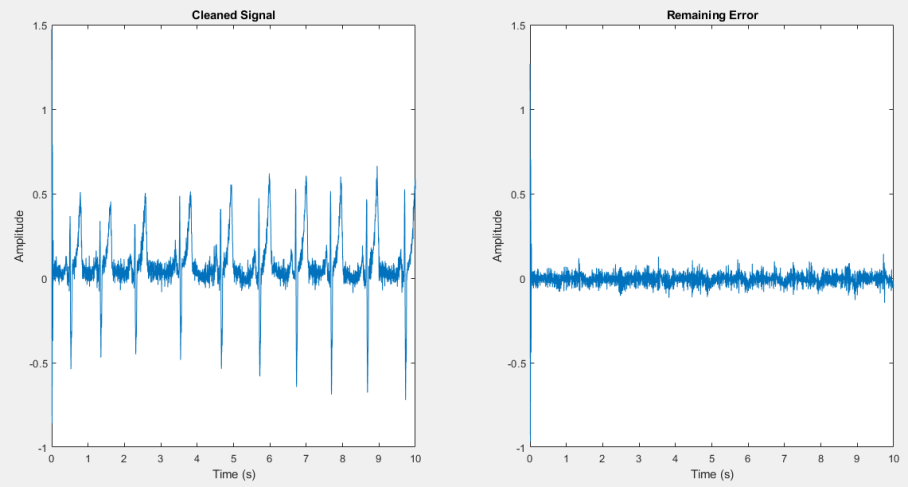

Figure 17: Cleaned Signal with Gain=0.01 and M=12 (left) vs. Remaining Error (right).

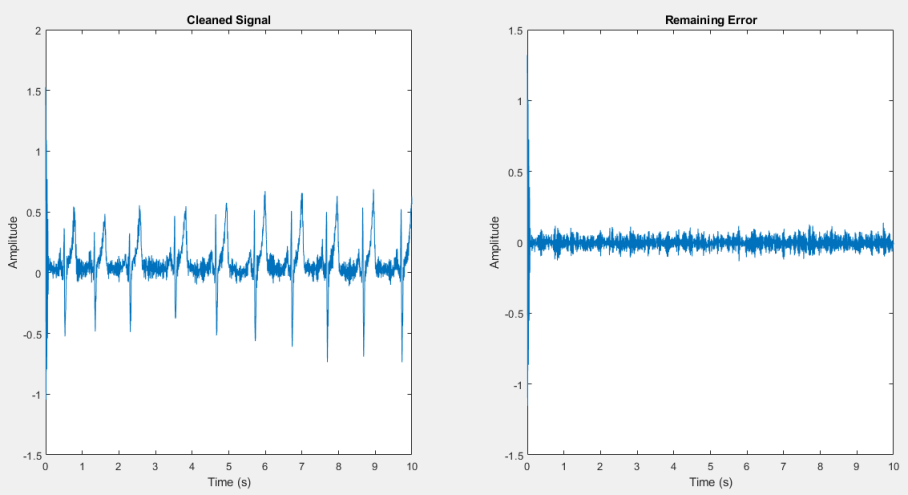

Figure 18: Cleaned Signal with Gain=0.01 and M=20 (left) vs. Remaining Error (right).

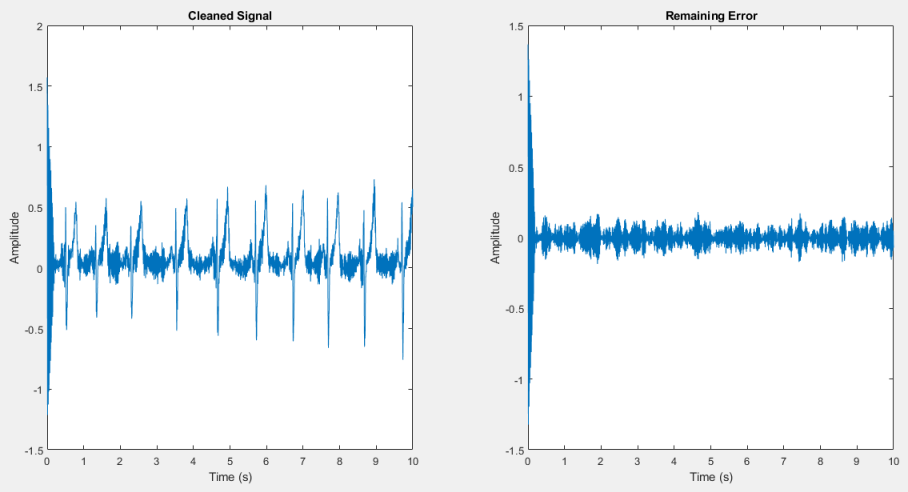

Figure 19: Cleaned Signal with Gain=0.01 and M=50 (left) vs. Remaining Error (right).

M	10	12	15	20	50
RMS	0.0327	0.0334	0.0352	0.0384	0.0549

Table 4: M and RMS values with gain=0.01

Finally, comparing the obtained results from Table 4 with the previous ones from Table 3, can be observed that the convenient value of M is even smaller as the gain has been increased. In this case, the convenient M value was set around 10.

In conclusion, it can be said that the bigger M is, the smaller the remaining error is without taking noise into account. Otherwise, if there is any disturbance on the signal, the bigger its gain is, the smaller will be the most convenient M value.

3.1.6

Adding a new weight function to the filter

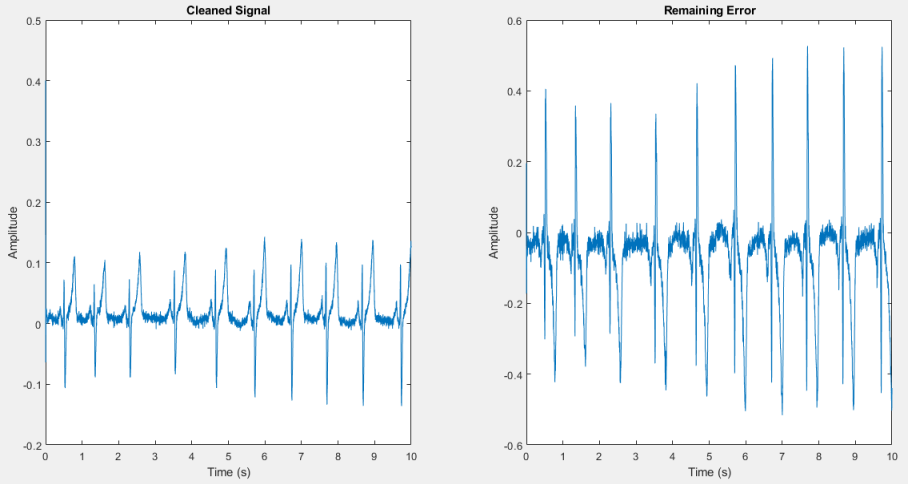

Figure 20: Cleaned Signal with Gain=0, lam=0.5, and M=100 (left) vs. Remaining Error (right).

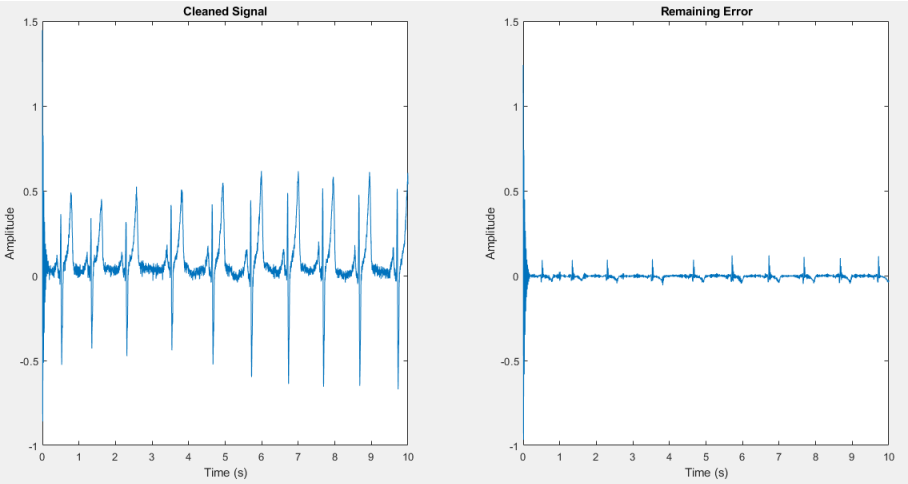

Figure 21: Cleaned Signal with Gain=0, lam=0.95, and M=100 (left) vs. Remaining Error (right).

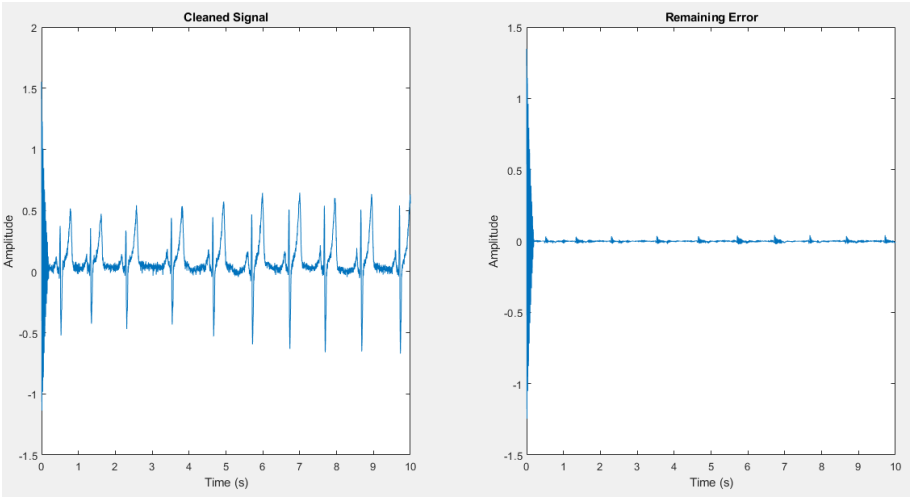

Figure 22: Cleaned Signal with Gain=0, lam=0.99, and M=100 (left) vs. Remaining Error (right).

First, the dependence on lam value was observed for the same gain and M value, as can be observed in the figures above. Clearly a decreasing tendency on the remaining error can be seen. Thus, the first conclusion is that the higher lam value is, the lower the remaining error is.

RMS	M=10	M=50	M=100	M=200
lam=0.5	0.1545	0.1545	0.1545	0.1545
lam=0.95	0.0244	0.0154	0.0153	0.0153
lam=0.99	0.0211	0.0066	0.0049	0.0043

Table 5: RMS values with gain=0 depending on M and lamda values.

Secondly, studying the dependence on M values, the Table 5 was compiled. At first sight, as observed on the previous tasks, it could be said that the bigger M value is, the smaller the remaining error is, but in this case this fact is not completely true.
Comparing the RMS values for the different M and lam values, can be seen that the remaining error stays independent on M when it gets big enough. The easy example is on the obtained values for lam=0.5, where can be seen that the RMS stays constant for all M values. In addition, for lam=0.95 and lam=0.99, as the M values are increased, the difference between the actual RMS value and the previous one gets thicker.
To sum up, implementing the weight function based on lam, on the one hand, it has been reported that a higher value of lam means a lower remaining error. On the other hand, a bigger M value also means a lower remaining error, but this error stays constant from a certain value of M depending on lam.
Ending the discussion of the second part of the report, if a comparison between the Figure 9 and 22 is done, can be observed that the main lobe generated on the remaining error around 0 is thicker applying the lam function. Otherwise, if Table 2 and Table 5 are compared, the RMS values are lower with the first weight function applied for the same gain and M value.

So, in order to determine which weight function should the user use, it would depend on their priorities. If a lower remaining error is desired, then the first function applied will be recommended. Otherwise, if the main lobe is wanted to be thicker, we should recommend the second weight function.