Group 41

2-2) tasks

2. Visual inspection

In this part the ECG signal for lead1 of all athletes have been plotted, and the heart rate for each case was estimated based on our visualization. Table 1 shows the results for all the athletes.

Table 1: Estimation of heartrate for lead 1 based on the plots of the data

Athlete	Heartrates/10 sec	Heartrates (bpm)
1	10	60
2	10	60
3	9	54
4	8.5	51
5	8	48
6	8	48
7	11	66
8	12	72
9	9	54
10	9	54
11	11	66
12	9	54
13	6.5	39
14	11	66
15	9	54
16	8	48
17	12	72
18	10	60
19	11	66
20	11	66
21	8	48
22	11	66
23	8	48
24	8	48
25	9	54
26	11	66
27	8	48
28	8.5	51

4. Heart rate Calculation

In this section, a function was defined in MATLAB. The inputs of the function were the ECG traces from the data and the sampling frequency (fs), the function calculates the heart rate of the given ECG, using Fast Fourier Transform. Table 2 demonstrates the result values for all 28 athletes and 12 leads. (The values are in bpm)

Table 2: The values of heartbeat for all cases based on MATLAB calculations

Athlete	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_10	L_11	L-12	Mean value
1	60	120	120	120	60	120	60	60	60	60	60	60	80
2	66	66	66	66	126	66	66	66	66	66	66	66	71
3	60	60	60	60	60	108	60	60	60	60	60	60	64
4	54	54	162	54	54	54	54	108	108	54	54	54	72
5	48	48	48	48	48	48	156	48	48	48	48	48	57
6	48	54	162	48	102	54	162	102	102	48	48	48	81
7	138	66	66	138	138	66	66	66	66	66	66	66	84
8	72	72	138	72	72	72	138	72	138	138	72	72	94
9	54	54	54	54	168	54	168	54	54	54	54	54	73
10	54	54	108	54	54	54	54	162	162	54	54	54	76
11	66	66	66	66	186	66	186	66	66	66	66	66	86
12	60	60	60	60	60	60	60	138	60	60	60	60	66
13	78	42	42	78	78	42	156	78	78	42	42	42	66
14	66	66	66	66	66	66	66	66	66	66	66	66	66
15	54	54	54	54	54	54	54	54	108	54	54	54	58
16	48	48	150	48	150	48	150	102	102	48	48	48	82
17	72	72	72	72	144	72	72	72	72	72	72	72	78
18	60	60	60	60	108	60	60	108	108	60	60	60	72
19	66	66	66	66	66	66	66	66	66	66	66	66	66
20	66	66	138	66	66	138	138	66	138	66	66	66	90
21	48	144	60	48	48	144	48	144	144	54	48	48	81
22	66	66	66	66	66	66	192	66	66	66	66	66	76
23	48	48	180	48	48	48	180	48	48	48	48	48	70
24	48	48	138	48	48	48	186	138	48	48	48	48	74
25	54	54	108	54	54	162	54	108	108	54	54	54	76
26	66	66	192	66	66	192	192	126	66	66	66	66	102
27	48	48	48	48	48	144	48	48	48	48	48	48	56
28	54	54	54	54	54	108	156	54	54	54	54	54	67

The mean values of the leads which was calculated by the code are slightly more than the visual inspections, but they still estimate the heart rate efficiently.

5. Results plot

Here is the comparison of the heart rate from visualization and calculation.

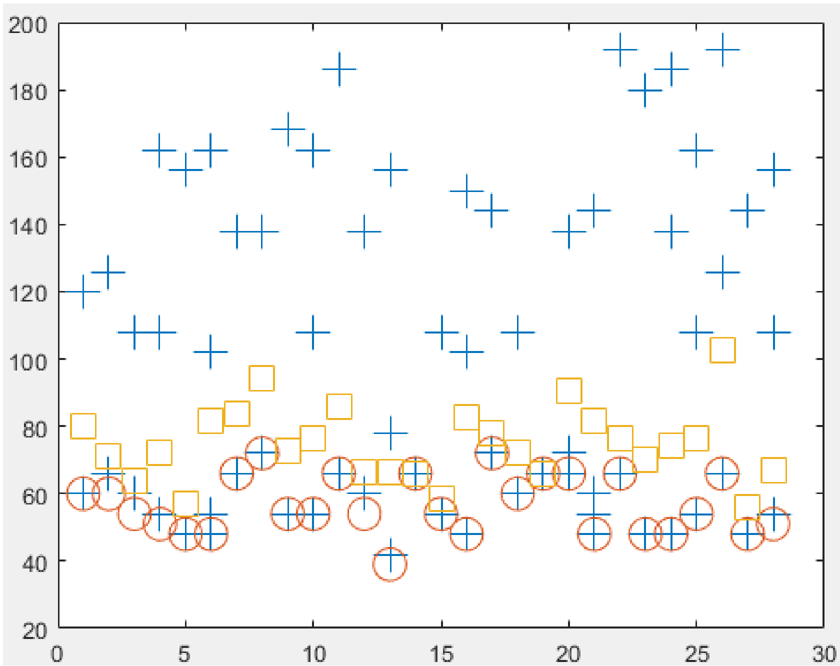

Figure 1: results plot

In the above graph, the x-axis is the athlete number, the and the y-axis is the estimate of the heart rate in BPM. The star '+' symbols illustrate the twelve estimates for each and every athlete as calculated by our matlab function fbpm. There is a star ploted for each and every lead and for each and every athlete, but due to the fact that for all leads we only have 2 or 3 possible values of bpm for the heart rate, the 12 stars for each athlete overlap each other. The heart rate from the visual inspection is represented by the 'o' symbol and the mean value from our 12 estimates is illustrated by a square.

The disrepancies between the estimated heart rate and the heart rate from the visual inspection have most probably to do with the fact that we used a very simple window function to carry out the matlab calculations.

2.3) Questions

1. We know that N_h denotes the number of samples we have for the data. In an equal time slot, with the increase of Nh, we can see that the F_s increases as well, and this makes the heart rate increase too.
2. The frequency grid is equidistant, and each grid corresponds to $\frac{2\pi f_s}{N_{dtft}}$. For a fft function which considers N samples, $x(n)\colon n = 0, \dots, N-1$, the frequency of the fft output would be $0, \dots, \frac{N-1}{N} 2\pi f_s$. Thus:

$$X(N_{dtft})\ would\ be\ at\ f = \frac{N_{dtft}-1}{N_{dtft}}2\pi f_s$$

3.

$$x(n) = \sum_{n=0}^{N_h} a_k\, cos(\omega_0 k\Delta t_n + \varphi_k)$$

$$\sum_{n=0}^{N_h} (a_k e^{j\omega_0 k\Delta t_n + \varphi_k)} + e^{-j(\omega_0 k\Delta t_n + \varphi_k)})/2$$

$$For\ n = 0 \rightarrow \frac{a_k}{2}\left(e^{j\varphi_0} + e^{-j\varphi_0}\right) = \beta_0$$

$$For\ n \geq 1 \rightarrow \sum_{n=1}^{N_h} a_k(e^{j\omega_0 k\Delta t}.e^{j\varphi_k} + e^{-j\omega_0 k\Delta t}.e^{-j\varphi_k})$$

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

3) Harmonic Disturbance Cancellation

3.1. Tasks

3. The following figures illustrate the use of the technique:

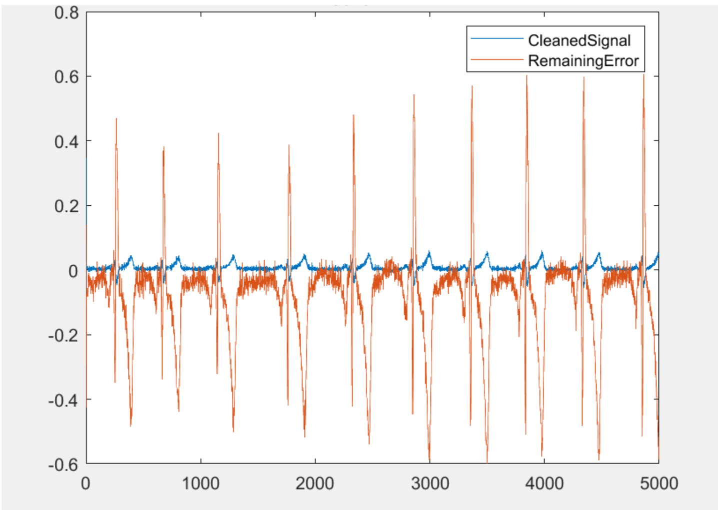

Figure 2. The resulting signal with gain=0, M=1

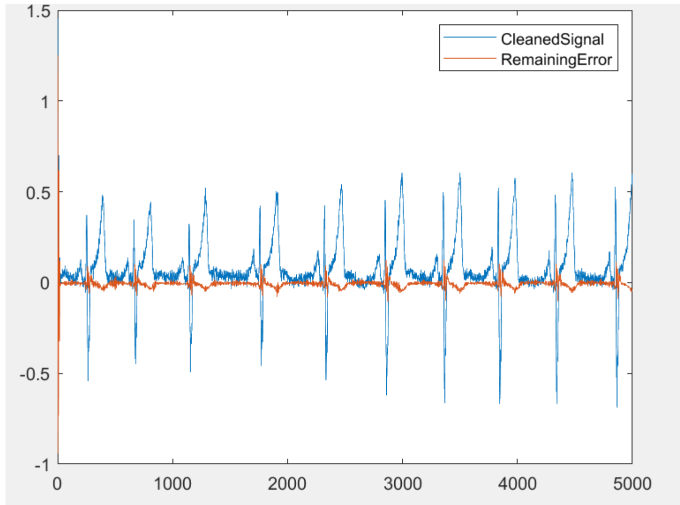

Figure 3. The resulting signal with gain=0, M=10

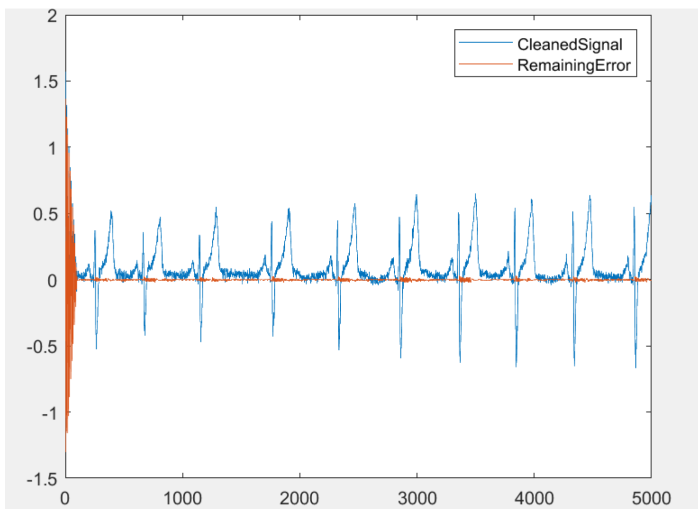

Figure 4. The resulting signal with gain=0, M=50

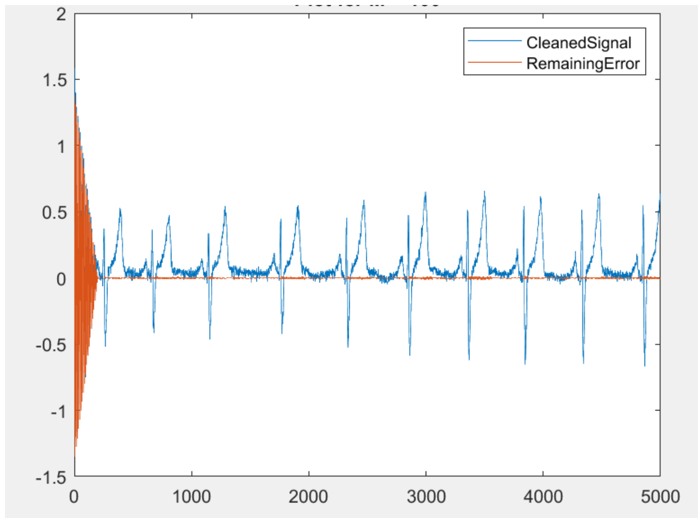

Figure 5. The resulting signal with gain=0, M=100

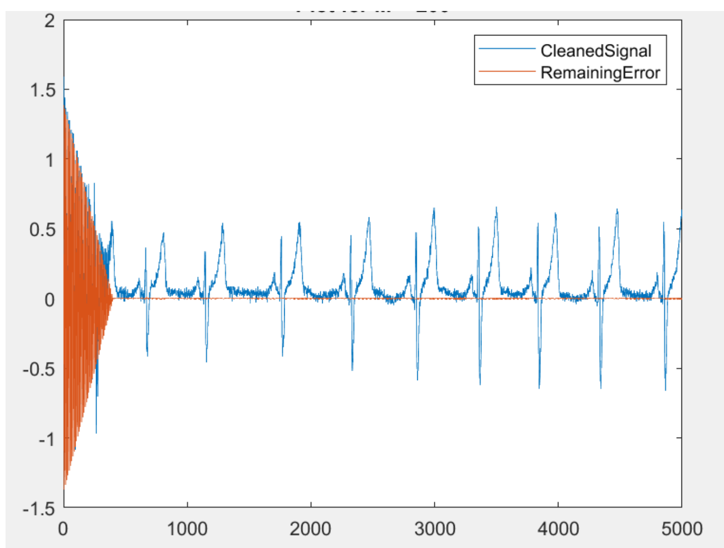

Figure 6. The resulting signal with gain=0, M=200

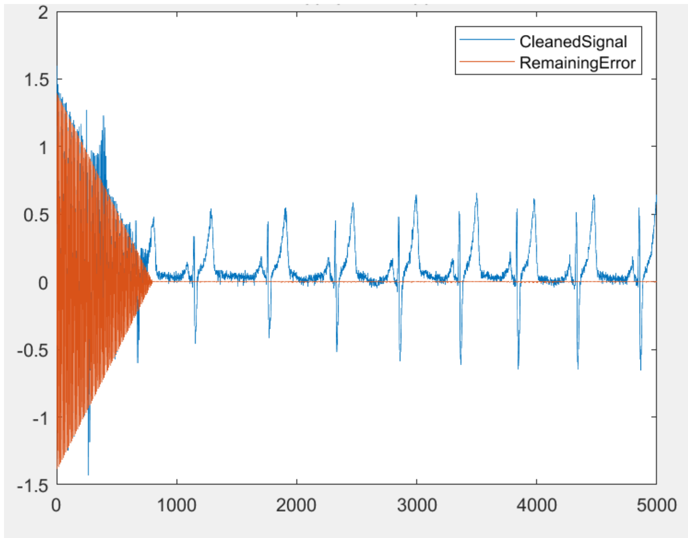

Figure 7. The resulting signal with gain=0, M=400

We observe that the best results are for M=10. For larger values a distrortion band is created which definitely affects the cleaned signal and will mess with our results if we try to analyse it further.

4. Herebelow we can see relevant plots for different values of M while gain = 0.001

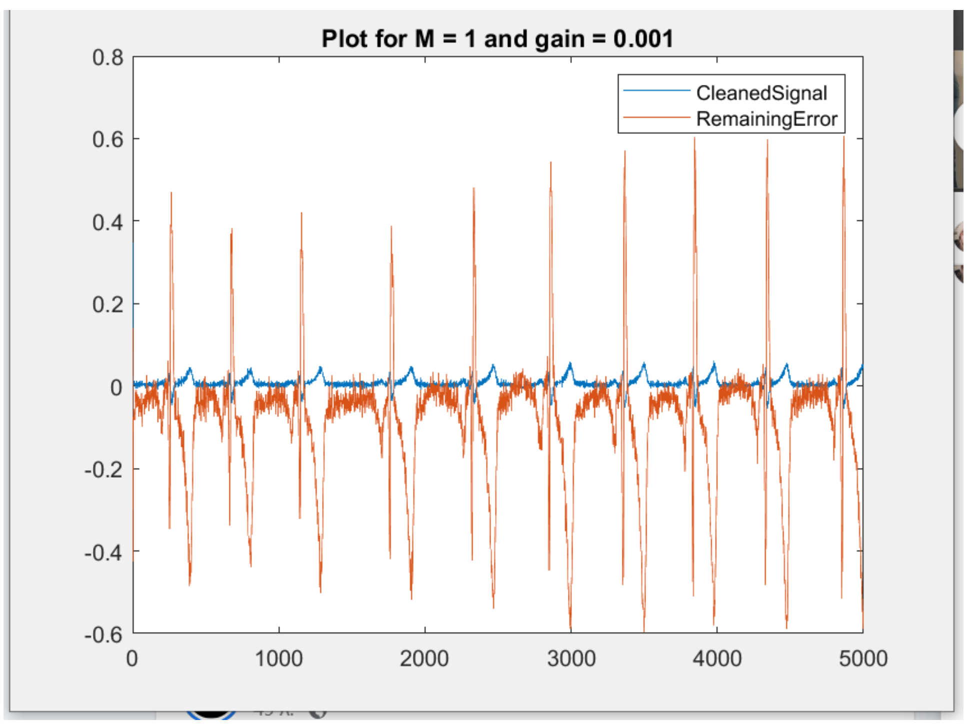

Figure 8. The resulting signal with gain=0.001, M=1

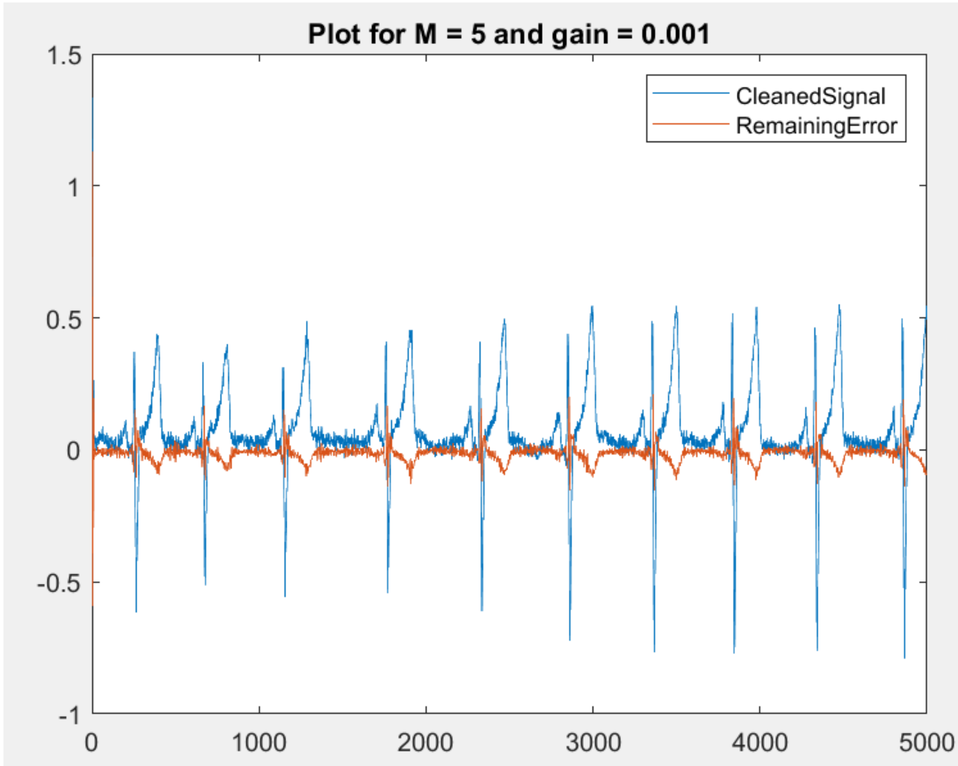

Figure 9. The resulting signal with gain=0.001, M=5

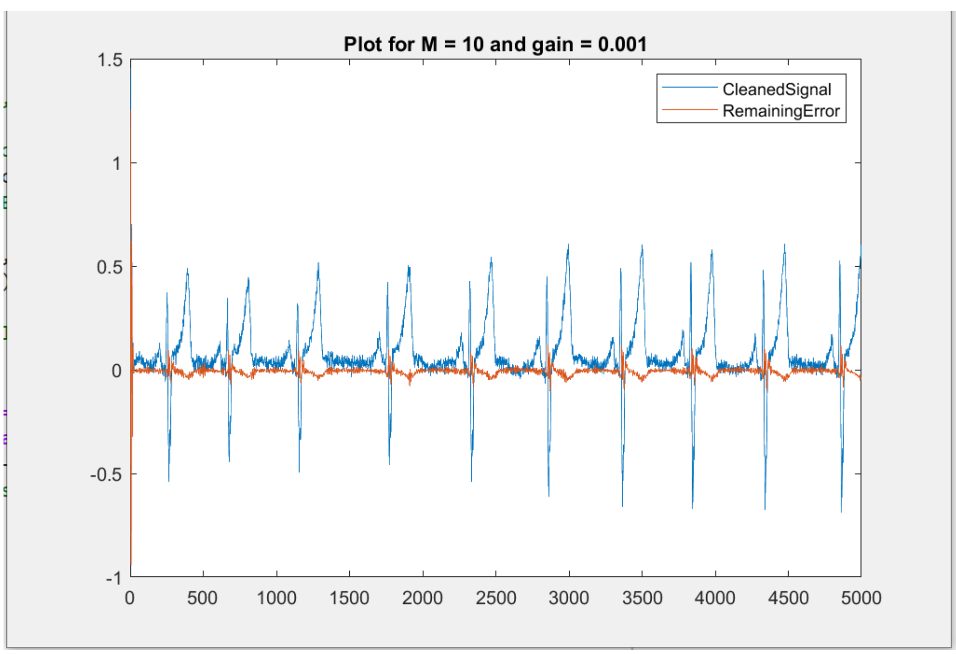

Figure 10. The resulting signal with gain=0.001, M=10

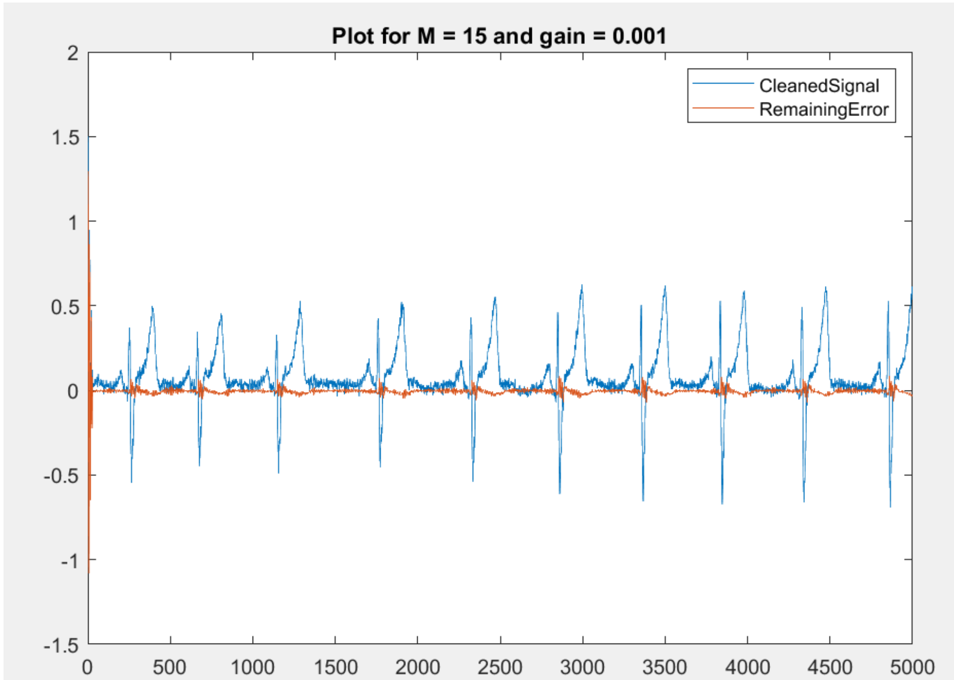

Figure 11. The resulting signal with gain=0.001, M=15

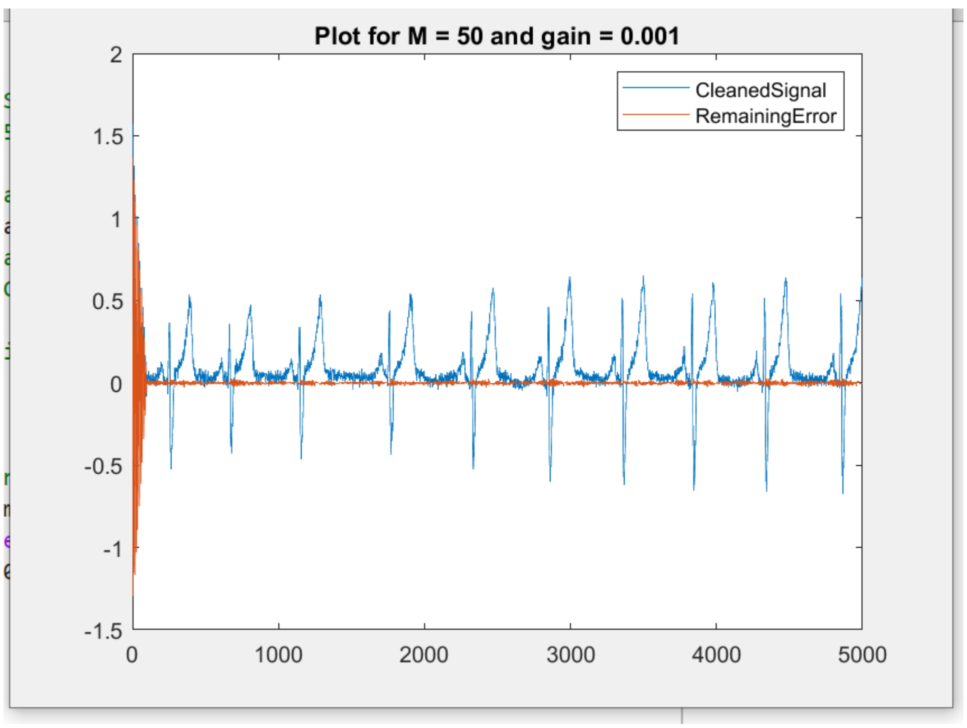

Figure 12. The resulting signal with gain=0.001, M=50

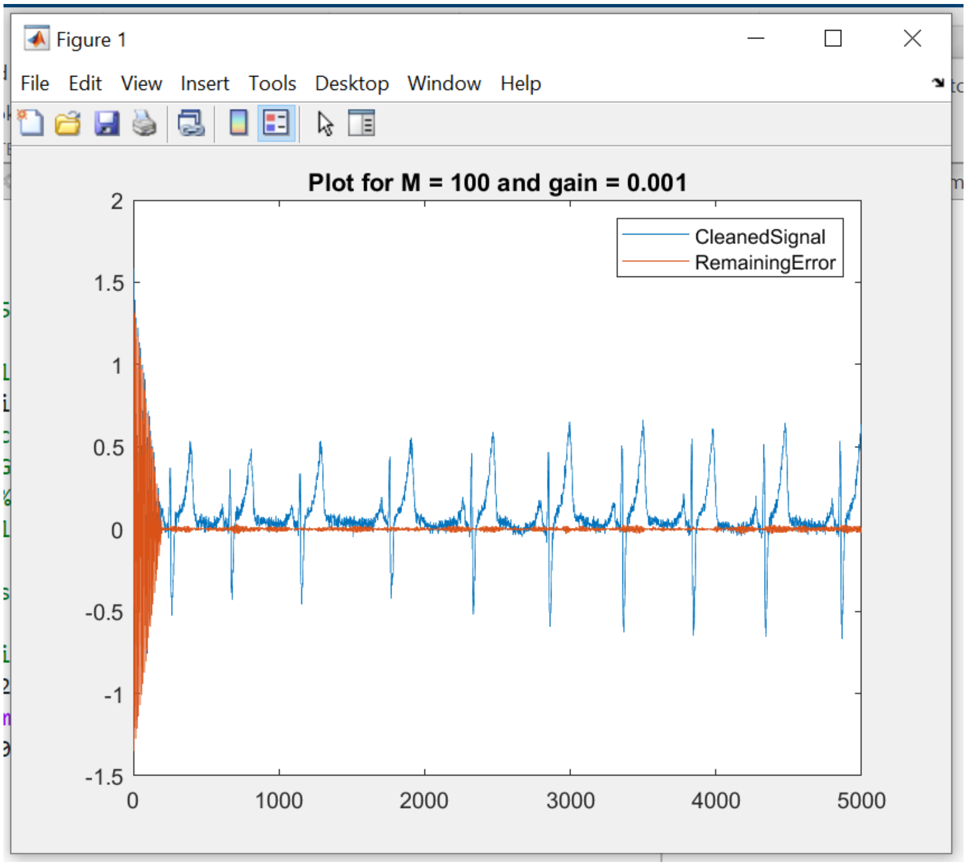

Figure 13. The resulting signal with gain=0.001, M=100

For the gain of 0.001, the best result seems to be for M=5 since same suffices without creating a distortion band. From M=5 to M=15 we get satisfactory results. However for larger Ms we get large distortion bands which greatly affect our signal and are undesirable.

5. Herebelow we can see relevant plots for different values of M while gain = 0.01

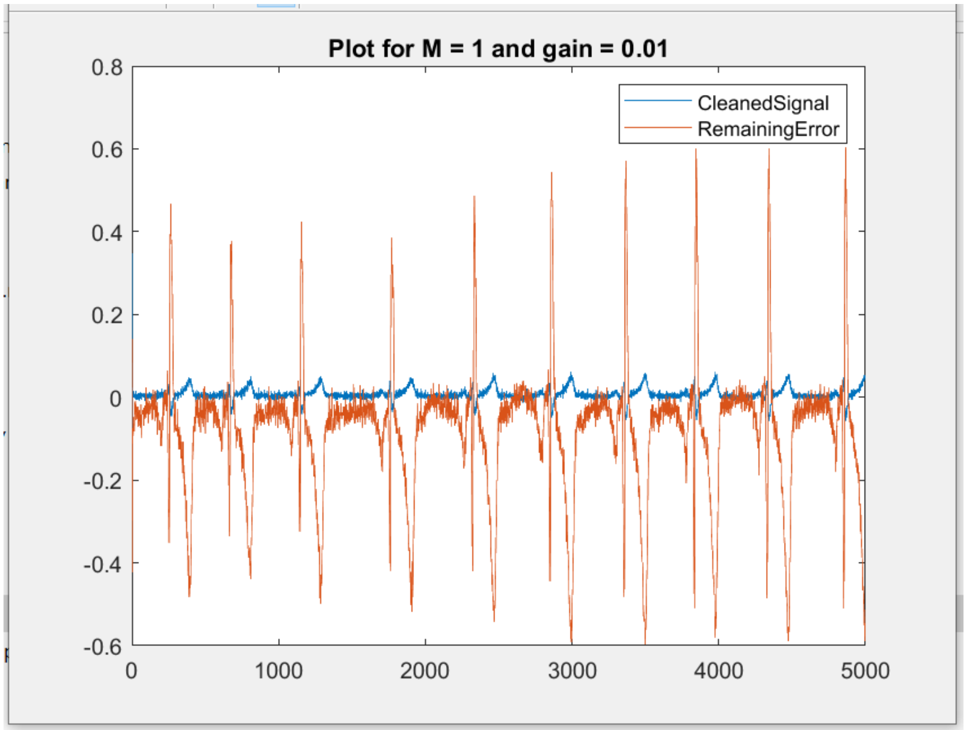

Figure 14. The resulting signal with gain=0.01, M=1

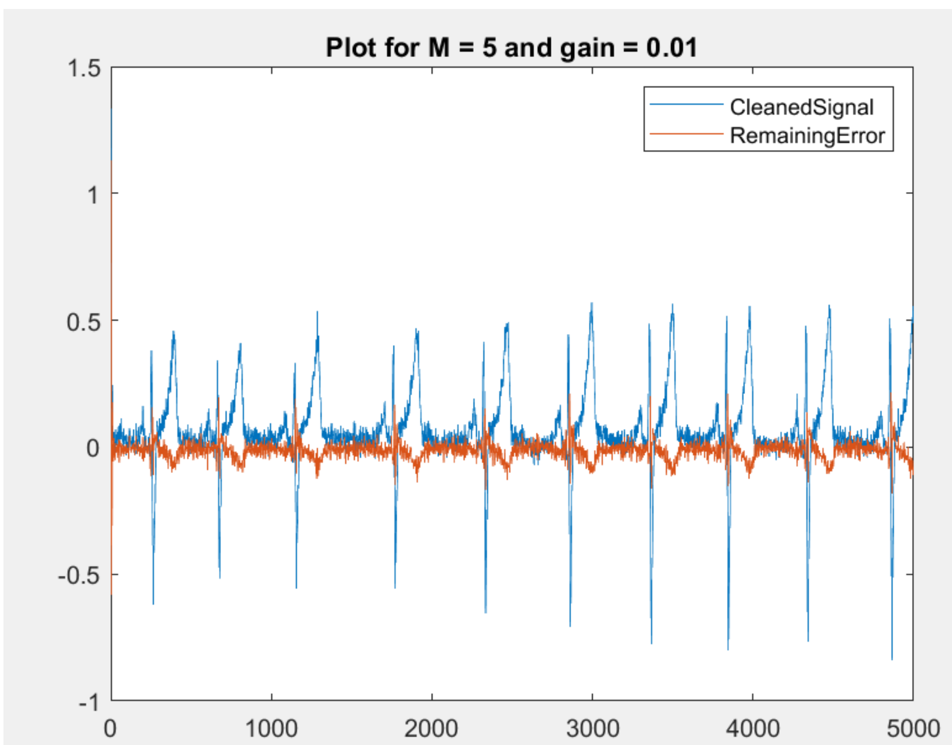

Figure 15. The resulting signal with gain=0.01, M=5

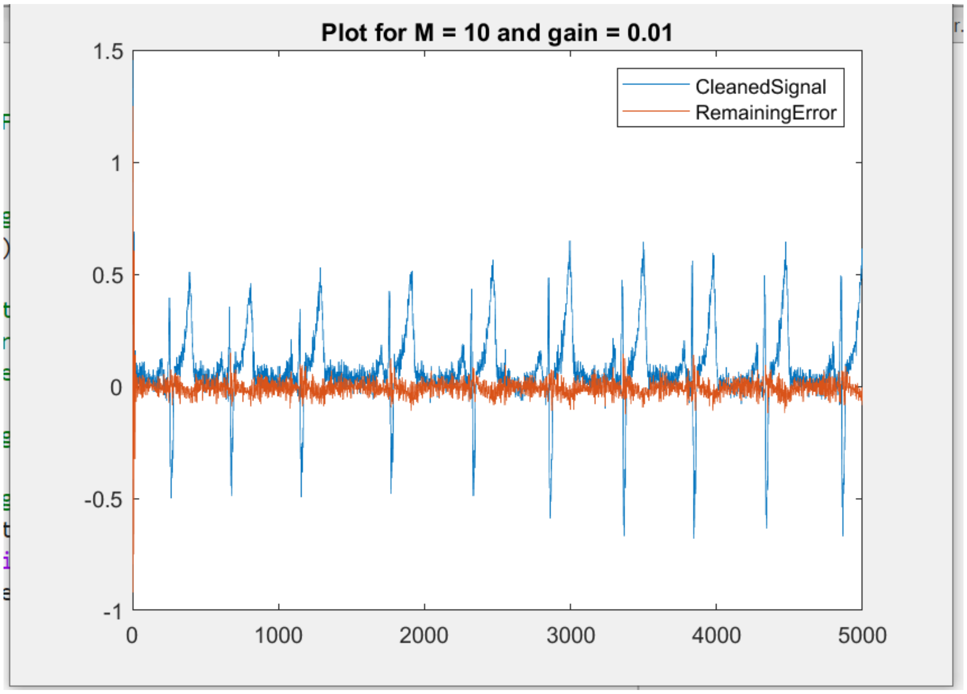

Figure 16. The resulting signal with gain=0.01, M=10

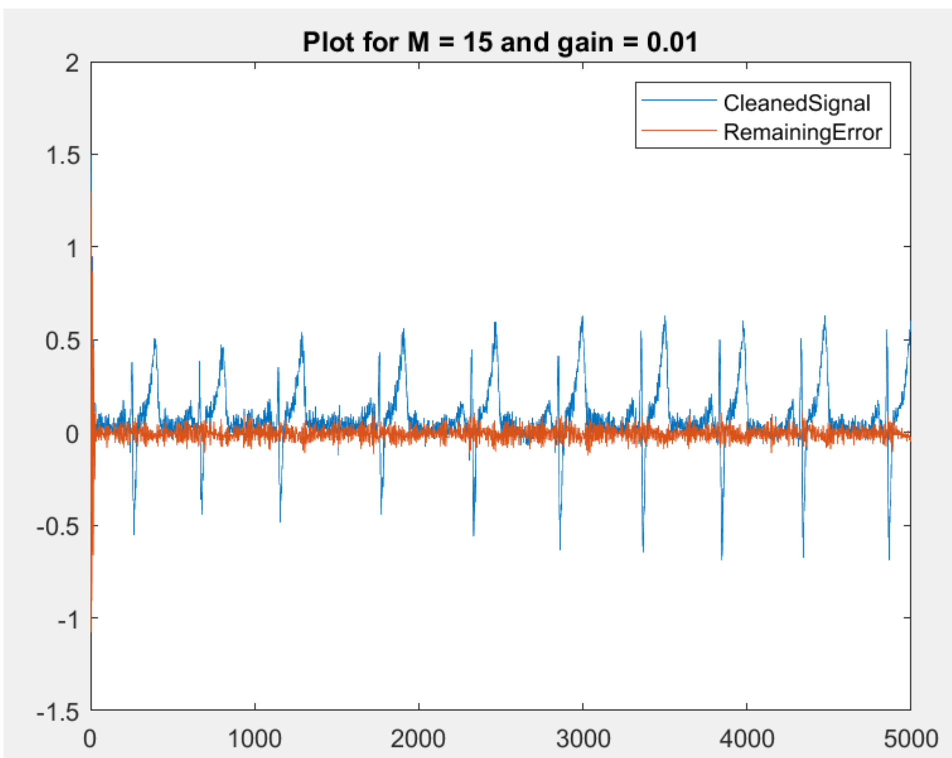

Figure 17. The resulting signal with gain=0.01, M=15

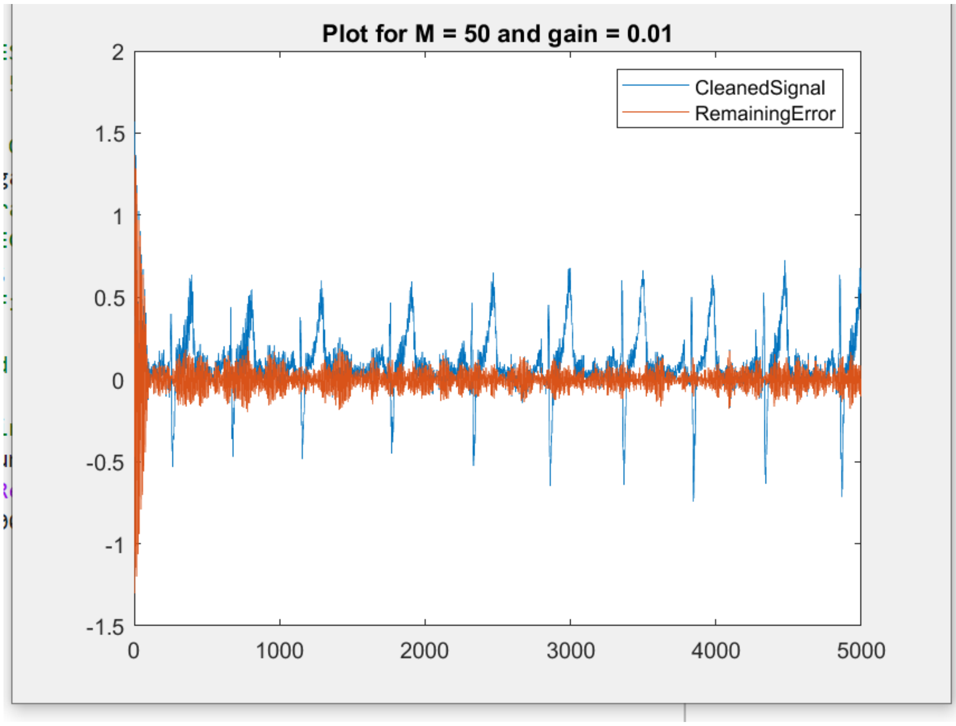

Figure 18. The resulting signal with gain=0.01, M=50

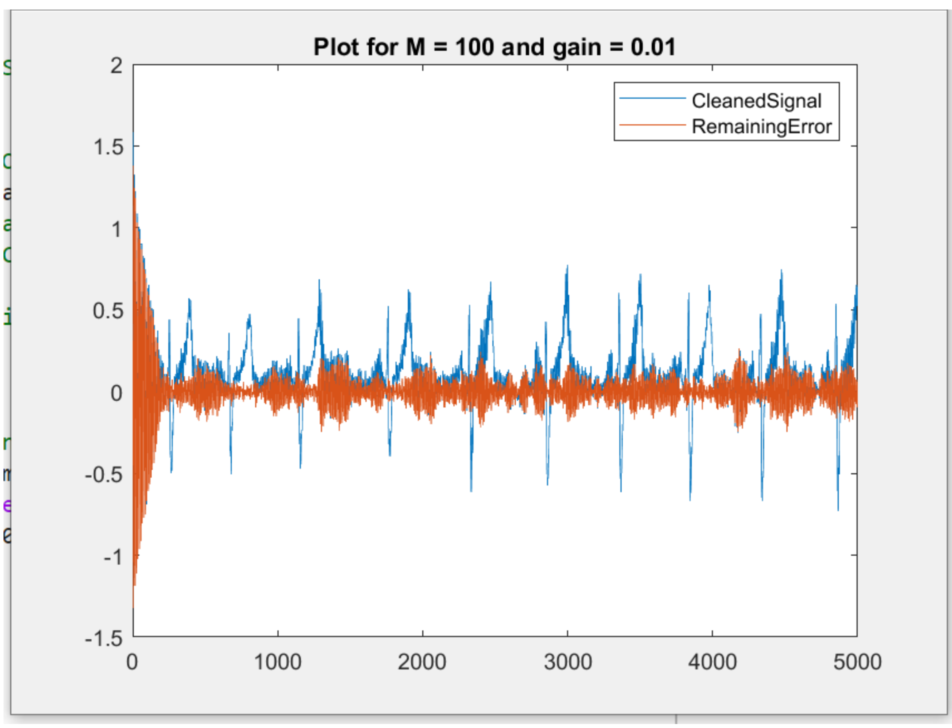

Figure 19. The resulting signal with gain=0.01, M=100

For the gain of 0.01, the best result seems to be for M=10. The optimal value of M changed and is greater compared to the previous case since we increased the gain of the distrurbance, which means that we introduced actually more noise to our signal, so M should be increased in order for our filter to work properly.

We also observe that as we increase our gain, the error in the results increases too (for values of M greater than 15).