SSY130 - Applied Signal Processing

Project 2: Adaptive Noise Cancellation

GROUP 41

GROUP MEMBERS: Polychronis Kanellos - Zhina Khalilollahi - Sanam Molaee - Seyedehnaghmeh Mosaddeghi

Empirical section:

QUESTION 1. The longest filter our implementation can handle is 200 taps.

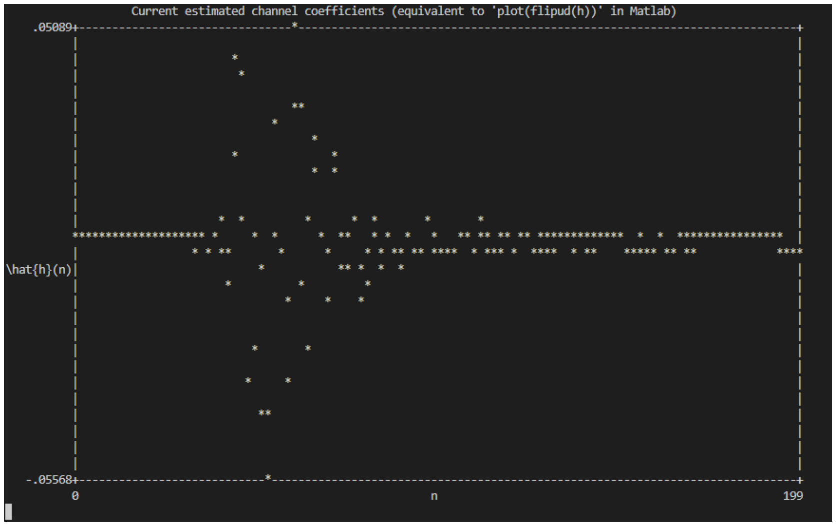

Figure 1. Filter Coefficients for Broadband Disturbance

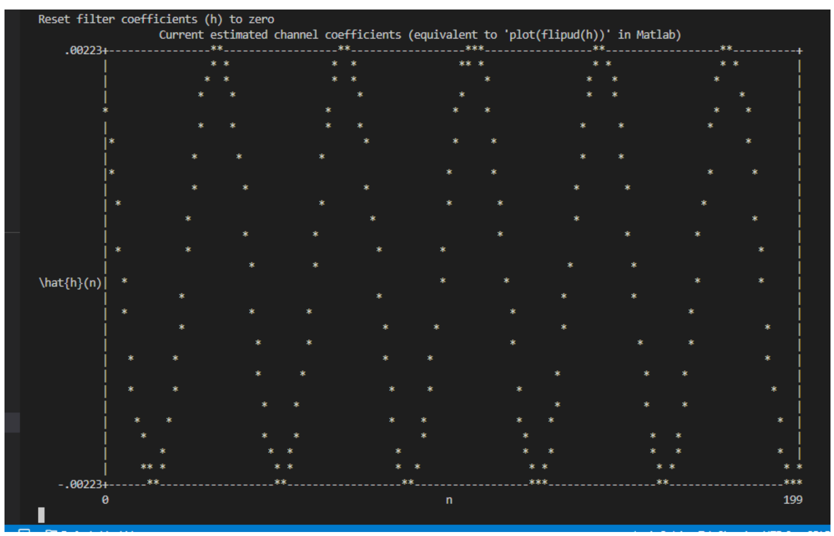

Figure 2. Filter Coefficients for Sinusoidal Disturbance

QUESTION 2.a)

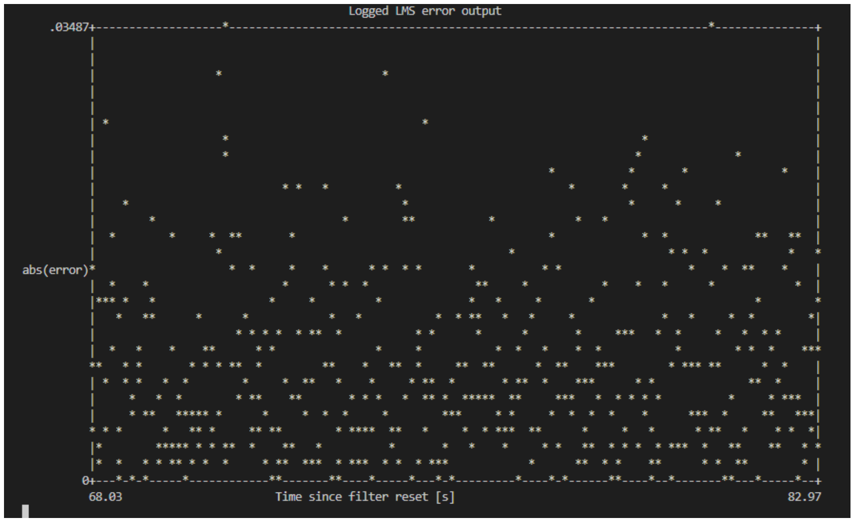

Figure 3. Input Signal: Broadband Noise – LMS Error with respect to Time for Original Setup (Without Channel Changes)

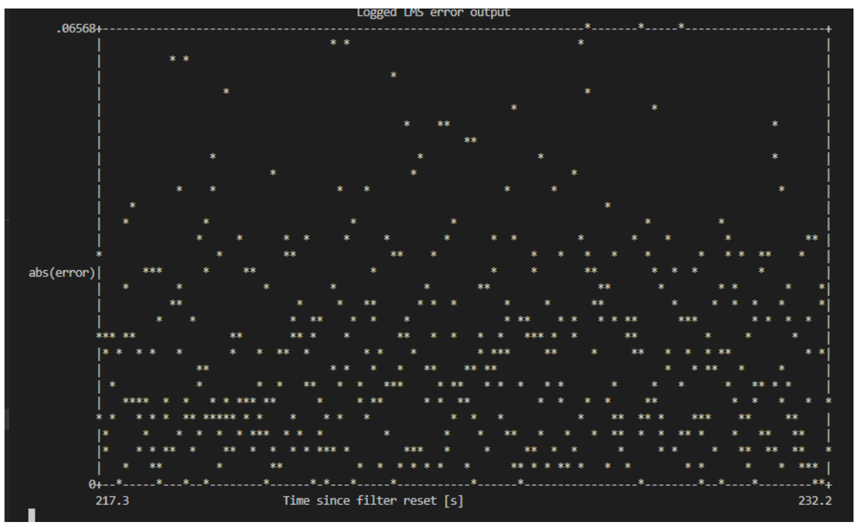

Figure 4. Input Signal: Broadband Noise – LMS Error with respect to Time after Moving the Disturbance (Signal Speaker) away from the DSP Microphone

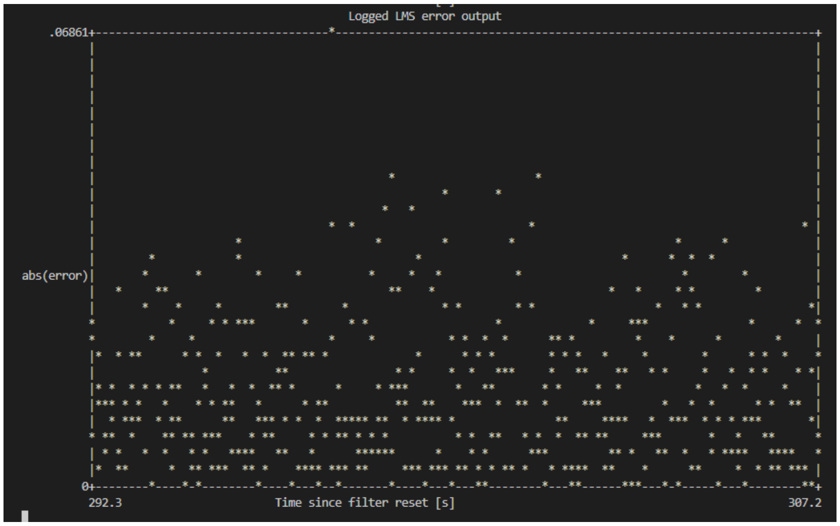

Figure 5. Input Signal: Broadband Noise – LMS Error with respect to Time for Obstacle Case (after placing a large book between the speaker and the DSP-kit)

After checking Figures 3-5 we can clearly see that for the BB Noise case when we change the physical channel (which is in our case air) by adding an additional disturbance of some kind (either by increasing the length of the physical channel, i.e. the distance of the Speaker from the DSP microphone or by adding an obstacle between the Speaker and the DSP microphone), the error output increases in amplitude approximately by a factor of 2. This happens due to the fact that the filter coefficients that are calculated for the above 3 cases are calculated for the initial channel only so when our physical channel changes they are no longer optimal, and thus the corresponding error increases.

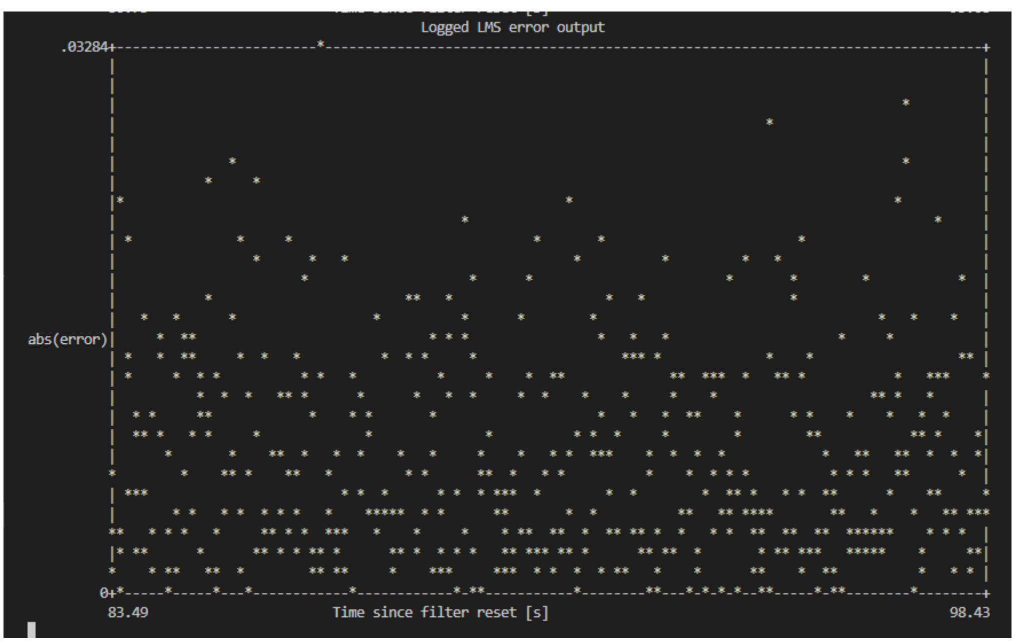

Figure 6. Input Signal: Sinusoidal Noise – LMS Error with respect to Time for Original Setup (Without Channel Changes)

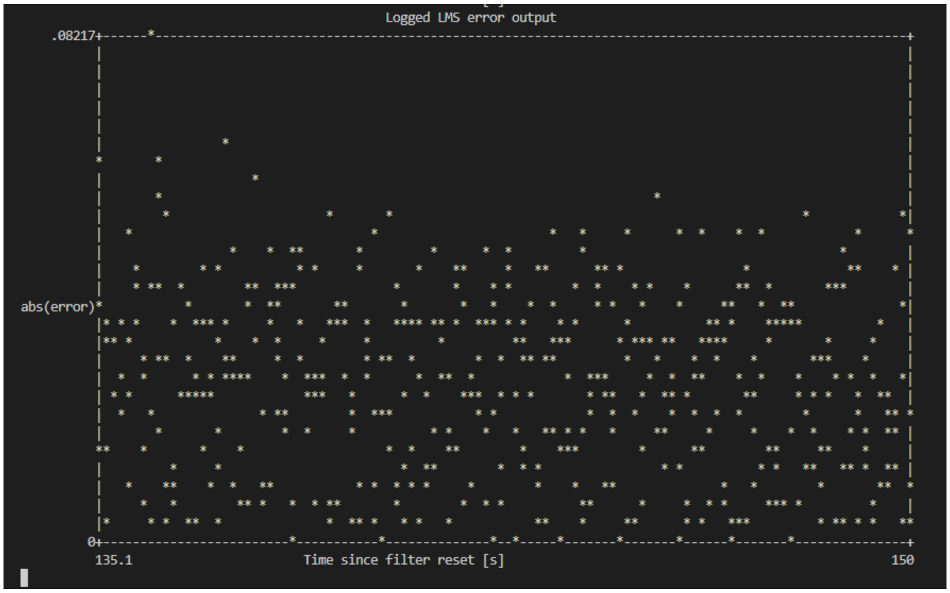

Figure 7. Input Signal: Sinusoidal Noise – LMS Error with respect to Time after Moving the Disturbance (Signal Speaker) away from the DSP Microphone

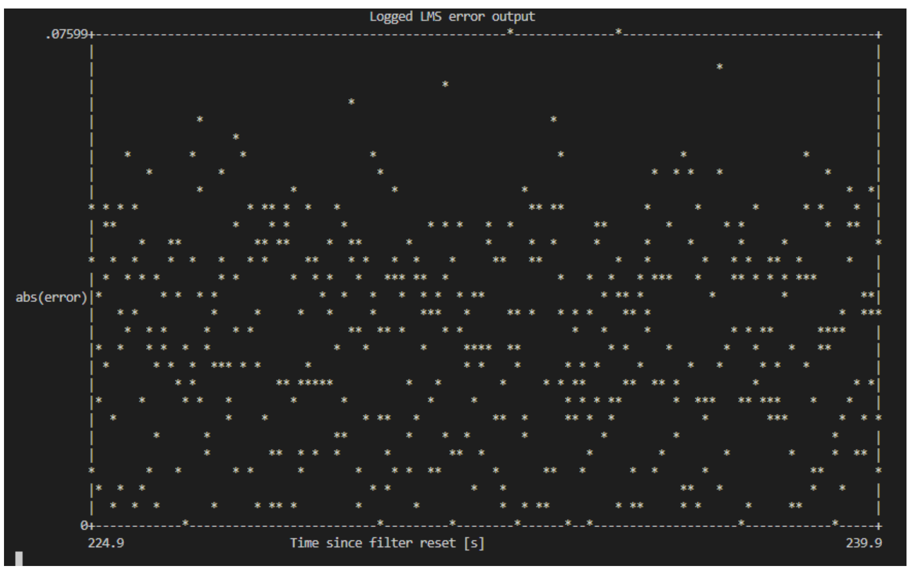

Figure 8. Input Signal: Sinusoidal Noise – LMS Error with respect to Time for Obstacle Case (after placing a large book between the speaker and the DSP-kit)

After checking Figures 6-8 we can see similar behavior for the Sinusoidal Noise Singal as we did for the Broadband Noise signal.

QUESTION 2.b) For the Broadband Noise case only, we started with an original level of speaker volume playing music and then we checked the following 2 cases:

(I) increasing volume for approximately 7 seconds and then decreasing it for approximately 7 seconds until we reached the original volume level

(II) decreasing volume for approximately 7 seconds and then increasing it for approximately 7 seconds until we reached the original volume level

The three stem plots for the above scenarios can be found in the following Figures 9-11.

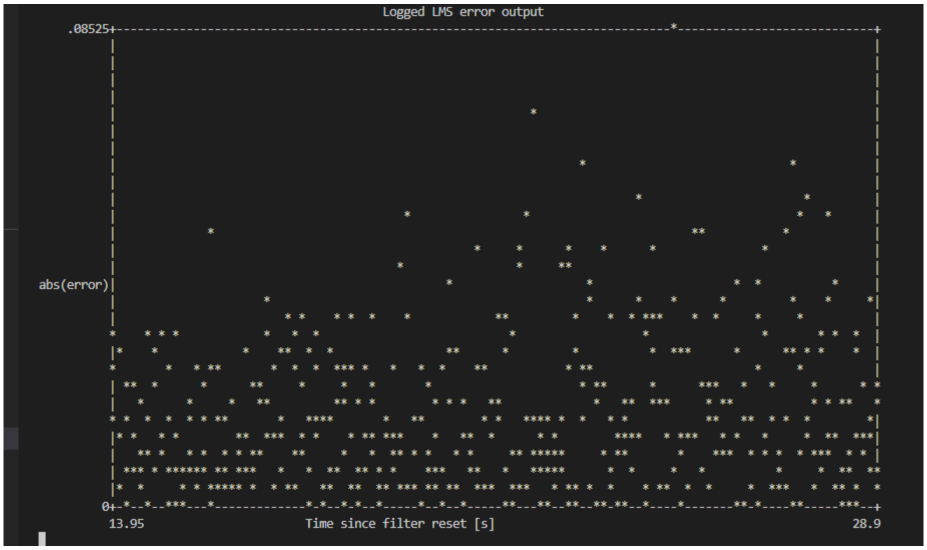

Figure 9. Input Signal: Broadband Noise – LMS Error with respect to Time for Original Speaker Volume

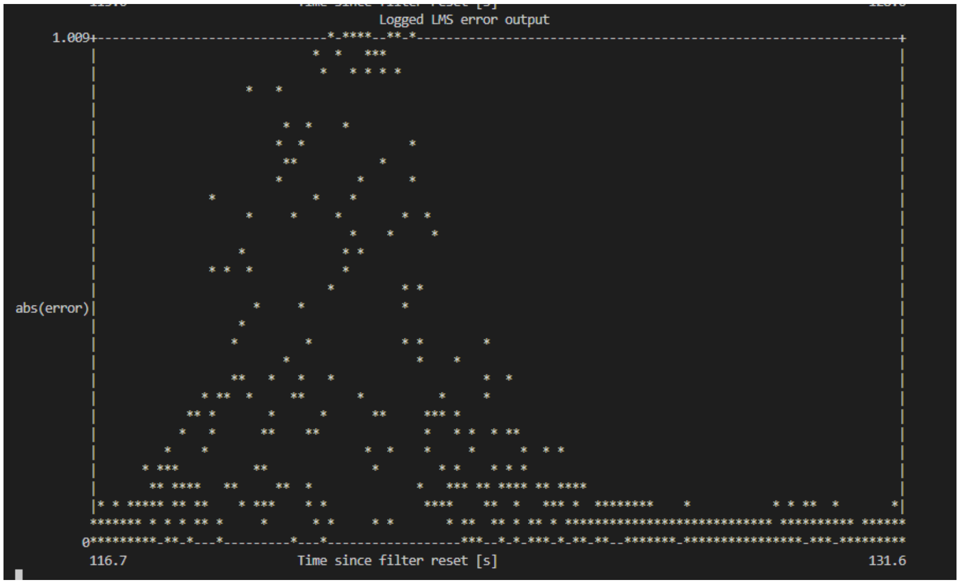

Figure 10. Input Signal: Broadband Noise – LMS Error with respect to Time for Scenario (I) - Raising Volume Scenario

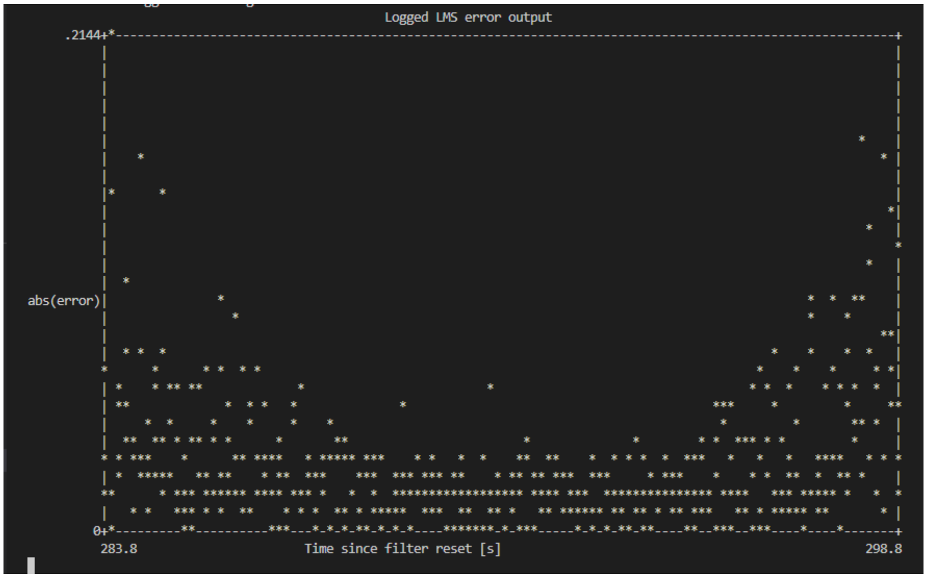

Figure 11. Input Signal: Broadband Noise – LMS Error with respect to Time for Scenario (II) - Decreasing Volume Scenario

In this case the error plots are clearly changed and we can see that for the Increasing and Decreasing Volume Scenarios the error is without a doubt bigger than the case (2a) for some time instances. Nevertheless, this does not mean that our filter is not working efficiently, because by increasing and decreasing the noise signal in this case we are also increasing and decreasing the amplitude of the desired signal as well which means that the calculated error gets affected as well. However, when we come back to the original volume the error returns to the same level as the case (2a) which means that our filter is still filtering efficiently the disturbance of our signal.

QUESTION 2.c)

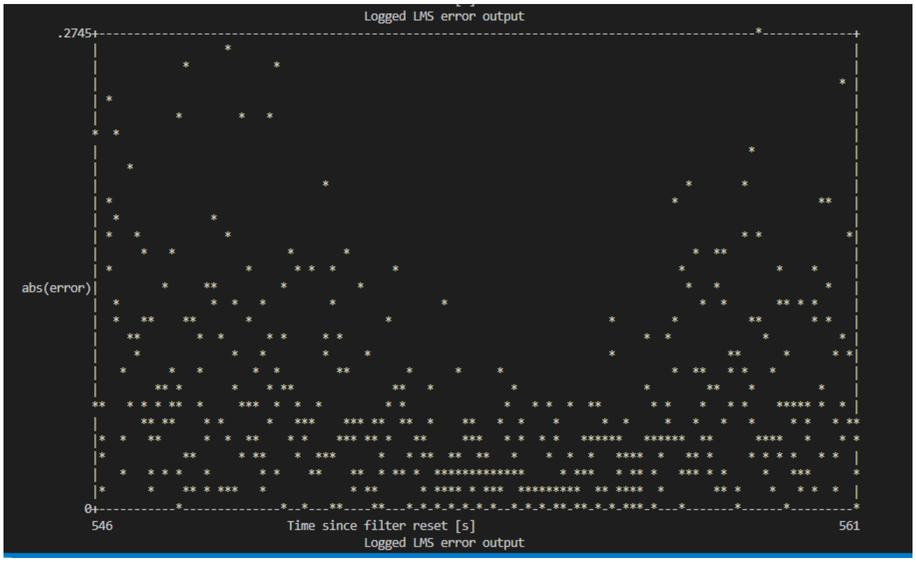

Figure 12. Signal Source: Cell Phone - Increase and then Decrease of Cell Phone Distance from the Microphone

The filter still works properly because when we gradually increased and decreased the distance from the microphone we see similar behavior as Figure 11 (a distance increase is equivalent to a volume decrease). When the cell phone was returned to its initial position close to the microphone the error levels were increased because the volume of the original signal was also increased which is expected behavior. The behavior is not the same as the case 2a. Because we now have not only a disturbance signal but also a music signal so we have larger error amplitudes.

QUESTION 3)

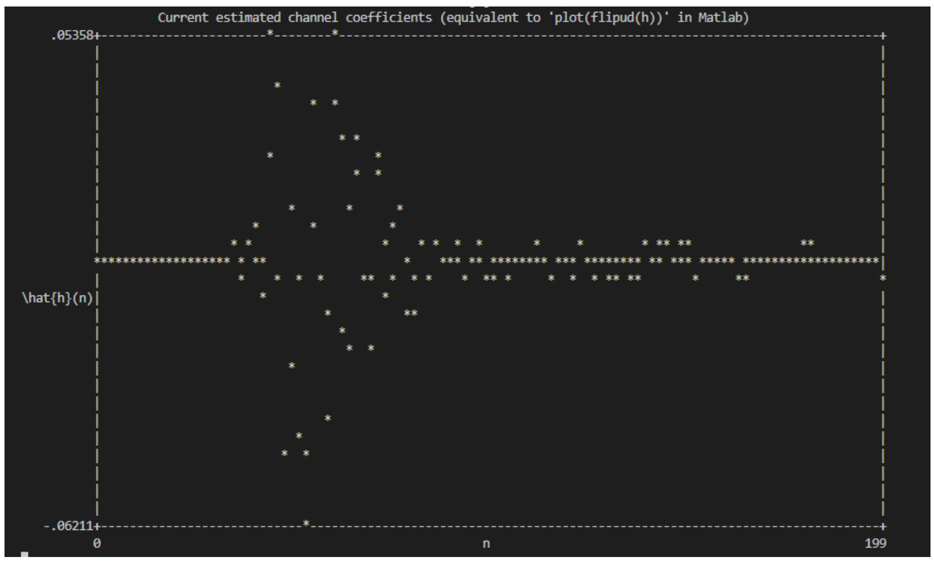

Figure 13. Stem plot of H_{bb} , 200 elements, no reset

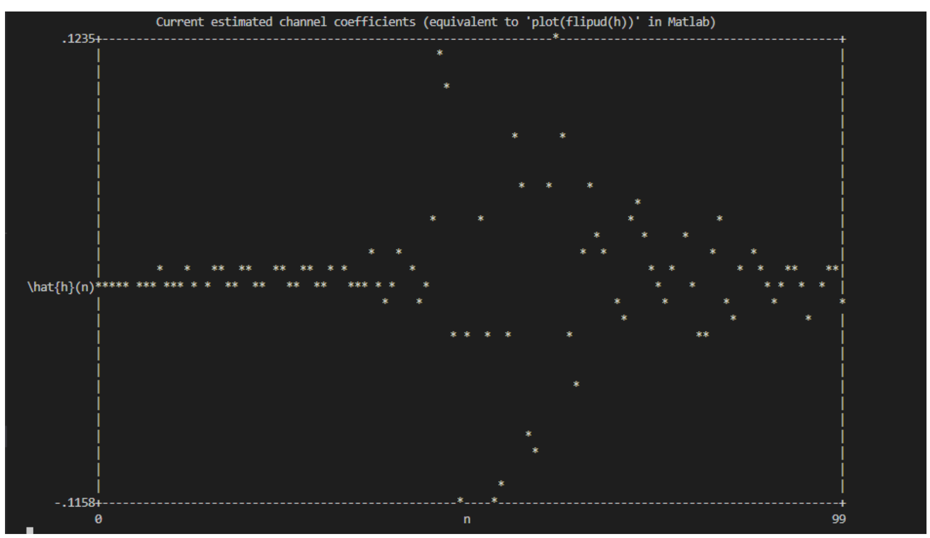

Figure 14. Stem plot of H_{bb} , 100 elements, no reset

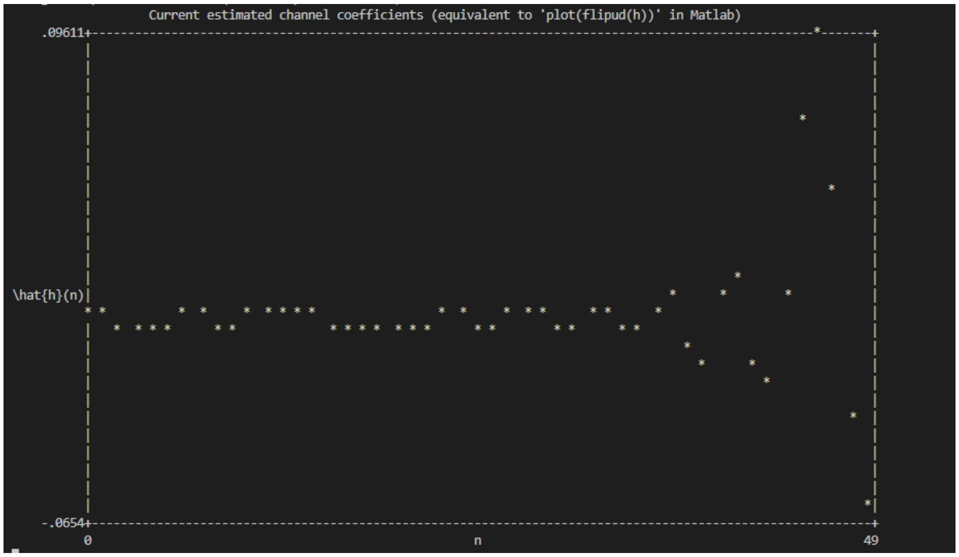

Figure 15. Stem plot of H_{bb} , 50 elements, no reset

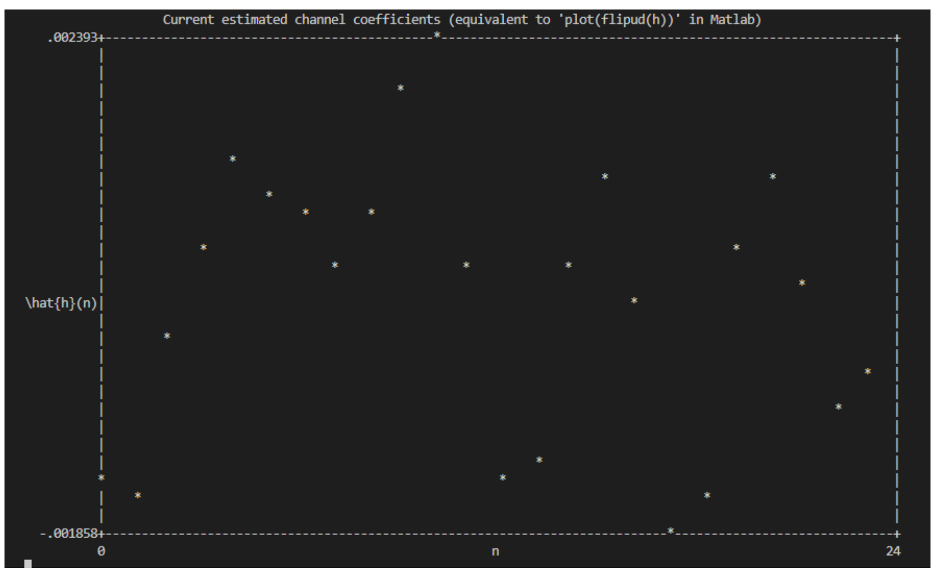

Figure 16. Stem plot of H_{bb} , 25 elements, no reset

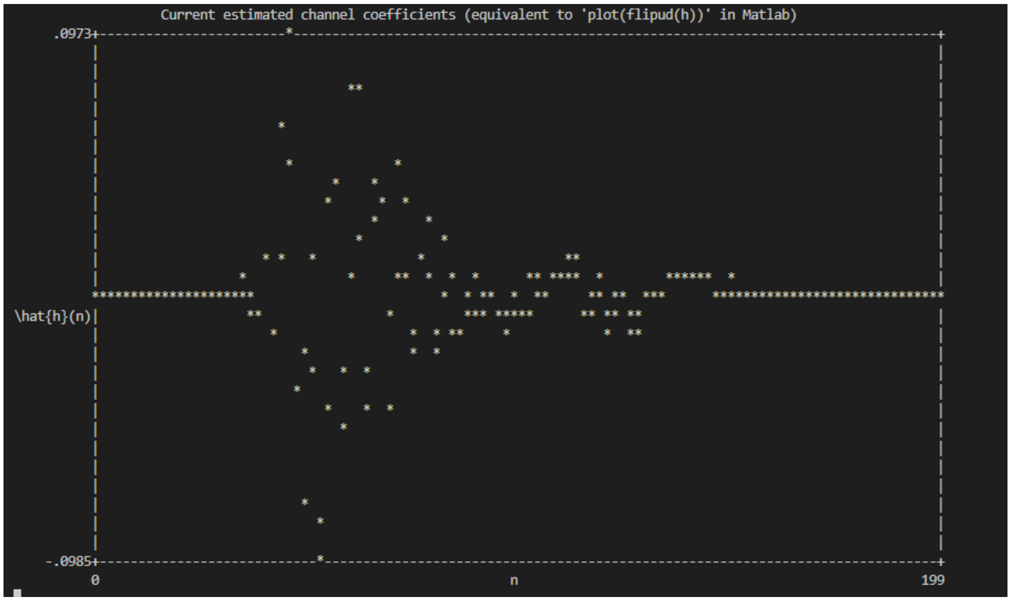

Figure 17. Stem plot of H_{bb} , 200 elements, reset

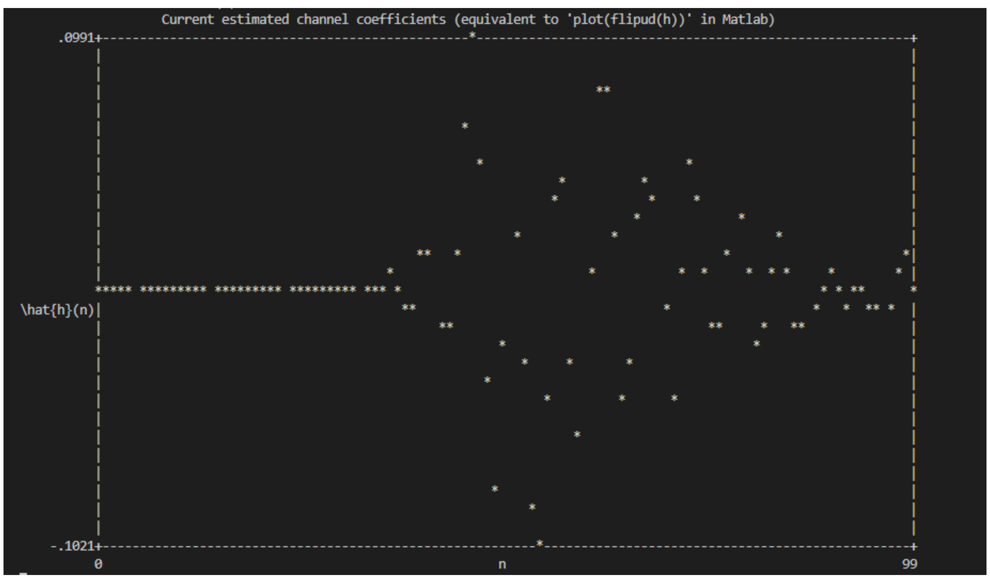

Figure 18. Stem plot of H_{bb} , 100 elements, reset

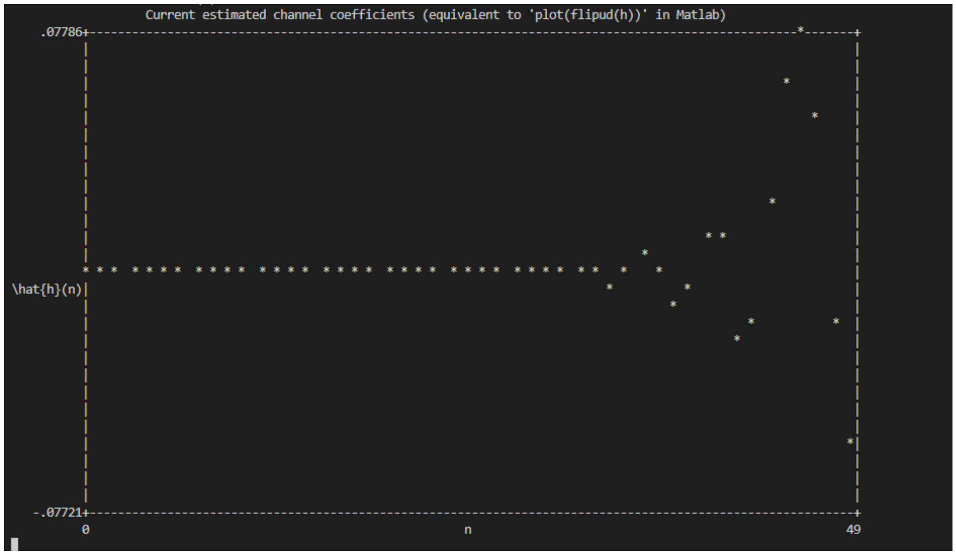

Figure 19. Stem plot of H_{bb} , 50 elements, reset

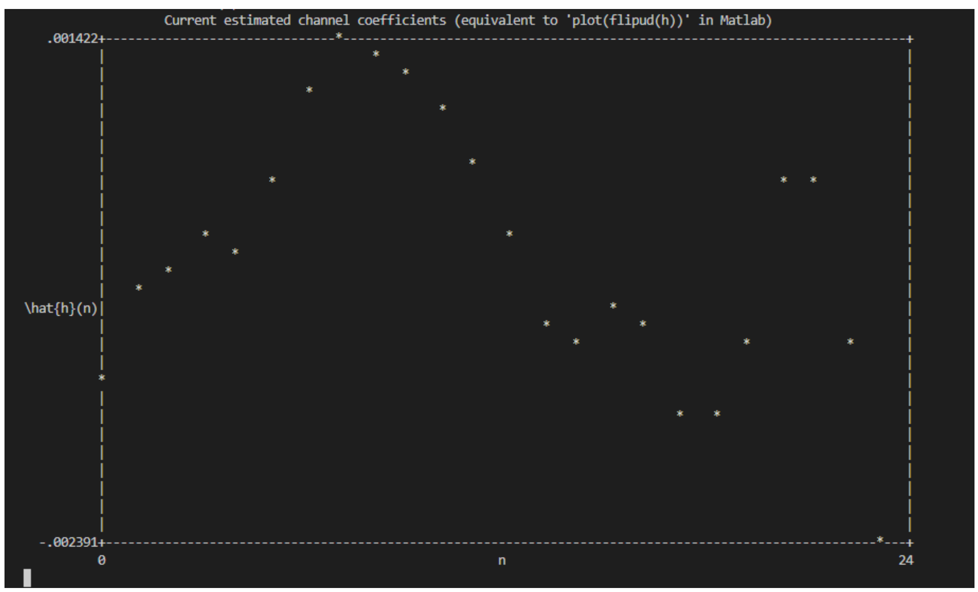

Figure 20. Stem plot of H_{bb} , 25 elements, reset

We can clearly see from the above Figures 13-20 that when we reset the filter coeffcients to zero after changing the filter length the filter has a harder time converging but it converges nonetheless. We can also see that the larger the length of elements is, the better our filter works. There is off course a critical length below which the performance becomes really poor. In the above plots we can clearly see that for lenghts below 50 the performance of our filter becomes really poor. When we decreased the number of elements experimentally and used the reset button to check the quality of our filtering process we found out that the actuall critical length was around 40 elements. By checking the stem plot of HBB of question 1 we can see that for number of elements n below 100 our calculated h starts losing convergence so we can assume that the critical length is below 100 elements.

QUESTION 4) The plots that show the requested changes are given here below:

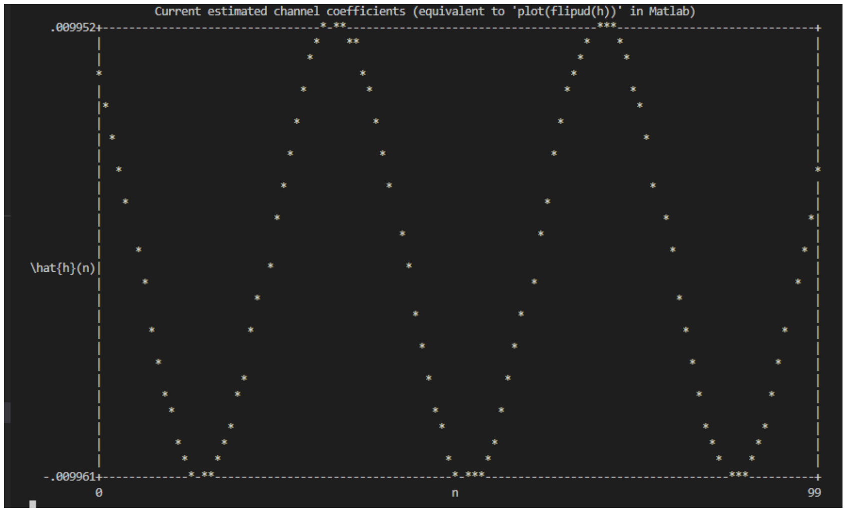

Figure 21. Stem plot of H_{sin}, 100 elements, no reset

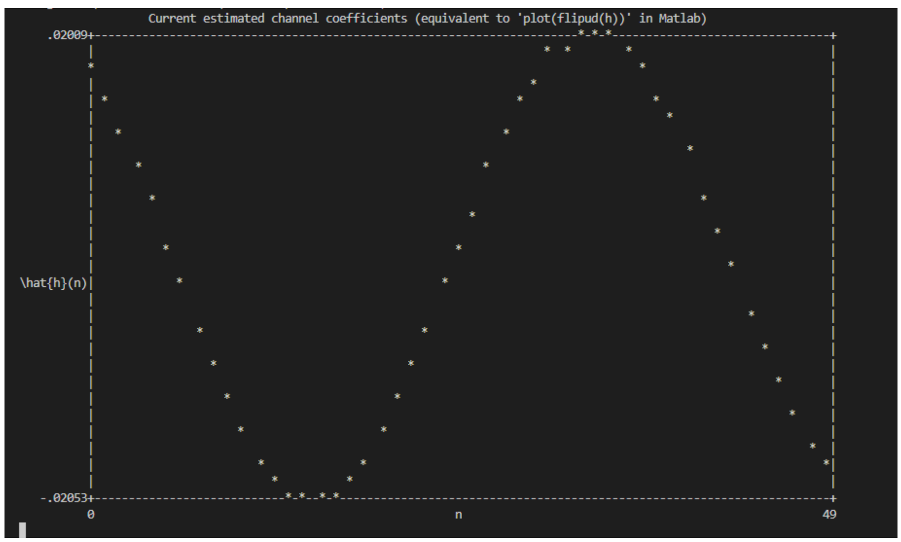

Figure 22. Stem plot of H_{sin}, 50 elements, volume decrease, no reset

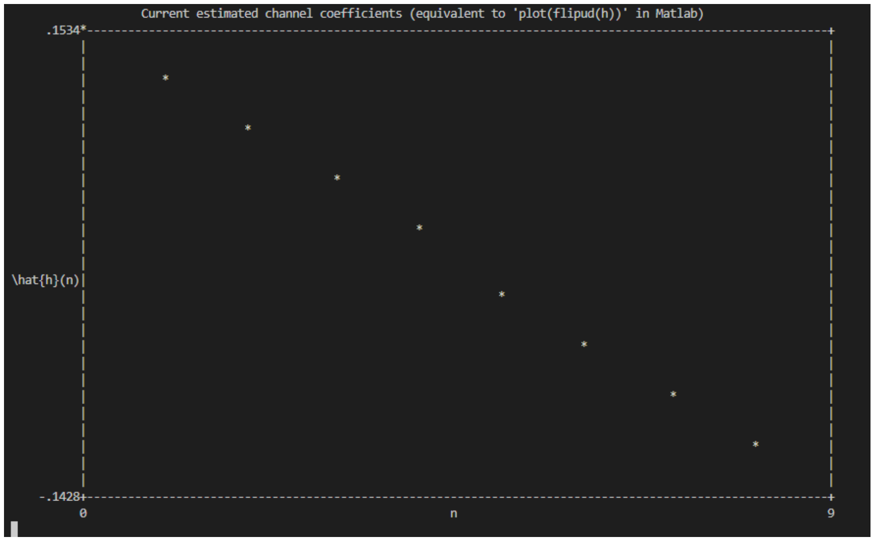

Figure 23. Stem plot of H_{sin}, 10 elements, volume decrease, no reset

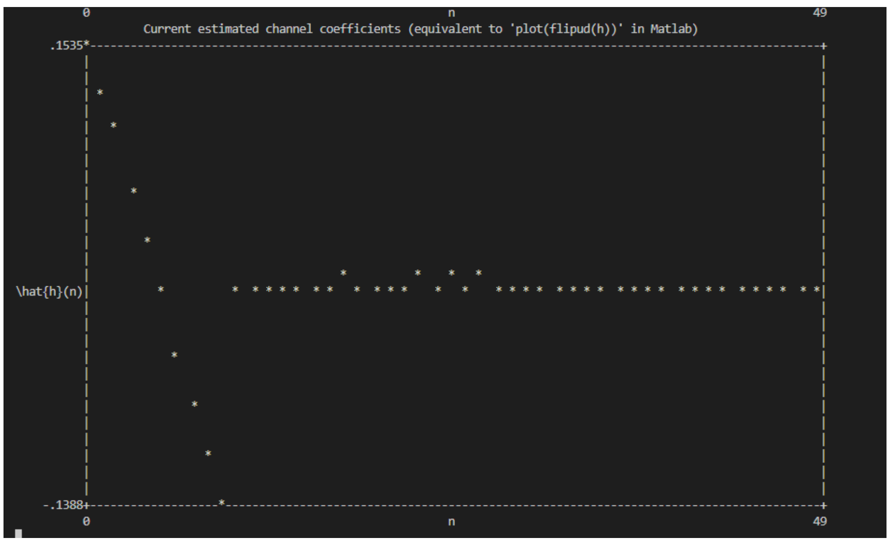

Figure 24. Stem plot of H_{sin}, 50, volume increase, no reset

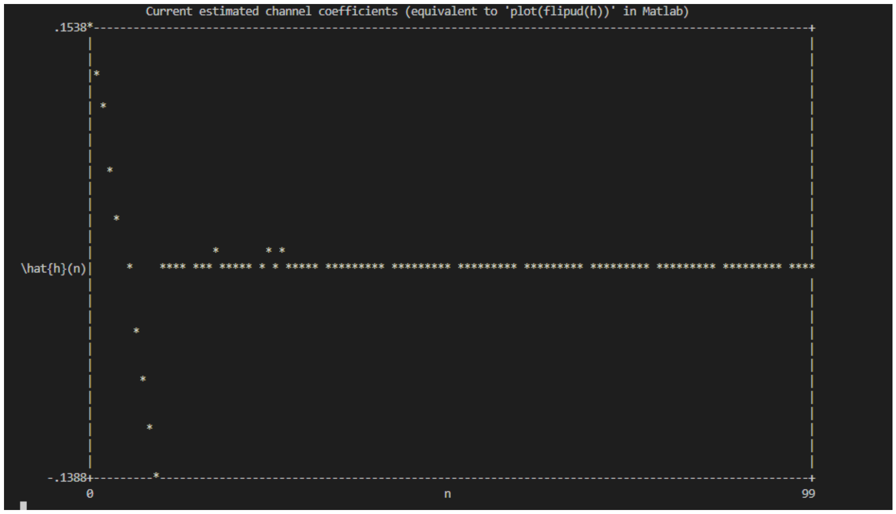

Figure 25. Stem plot of H_{sin}, 100 elements, volume increase, no reset

Figure 26. Stem plot of H_{sin}, 100 elements, volume decrease - after reset

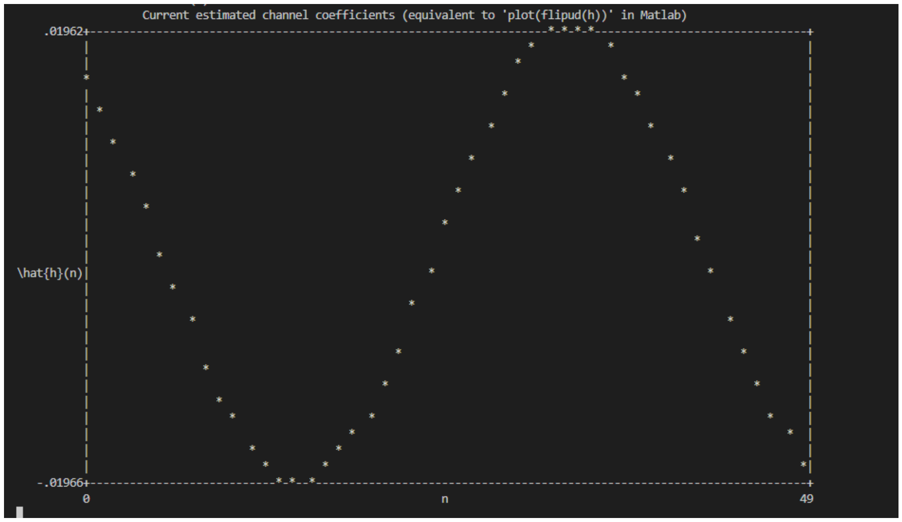

Figure 27. Stem plot of H_{sin}, 50 elements, volume decrease, reset

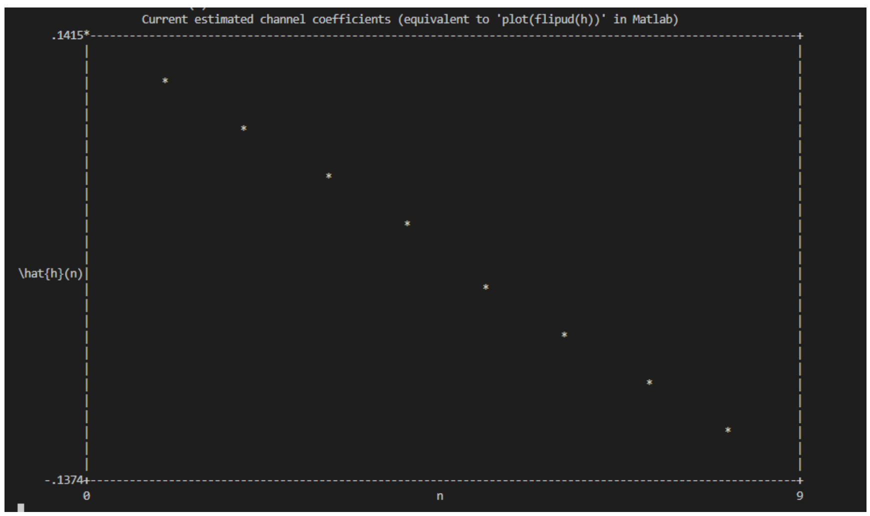

Figure 28. Stem plot of H_{sin}, 10 elements, volume decrease, reset

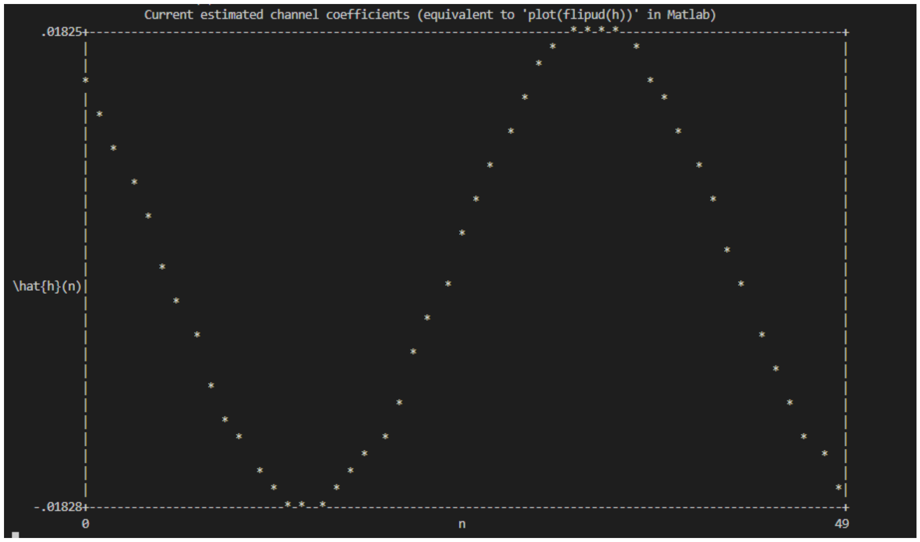

Figure 29. Stem plot of H_{sin}, 50 elements, volume increase, reset

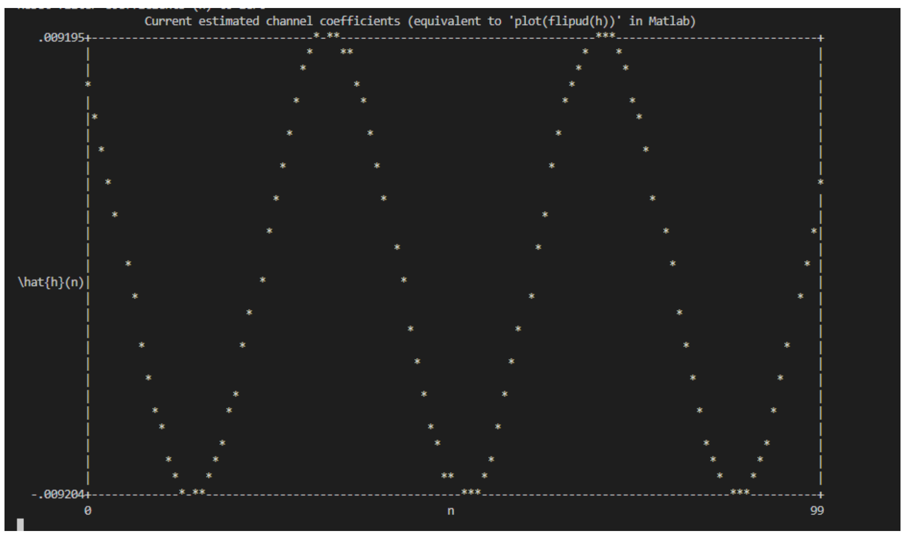

Figure 30. Stem plot of H_{sin}, 100 elements, volume increase, reset

The above results clearly depend on whether the filter coefficients are reset after changing the filter length due to the sinusoidal properties of the distrubance and due to the fact that the impulse response hbb is effected by previous channel inputs so if the channel is not reset then the hbb will be affected by the previous inputs. We can also clearly observe that when the filter length decreases so does the filter quality.

QUESTION 5) The requested plots can be found here below:

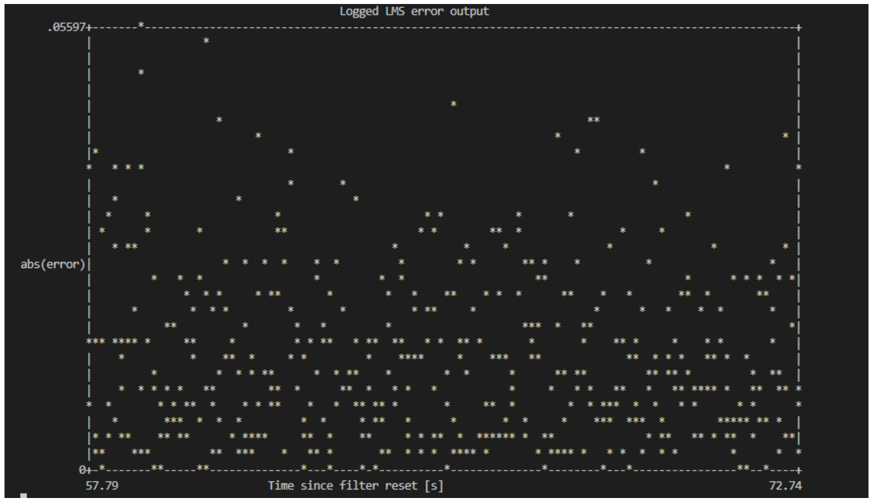

Figure 31. h_{sin} filter error for the sinusoidal disturbance

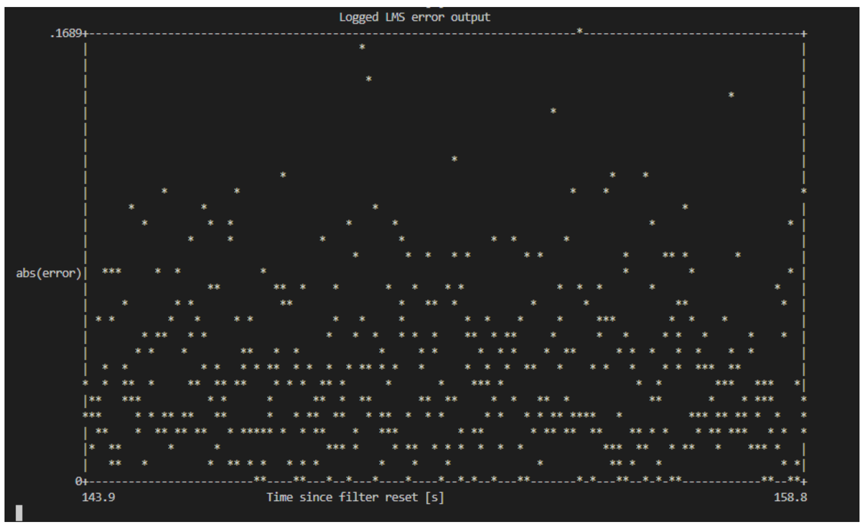

Figure 32. h_{sin} filter error for the broadband disturbance

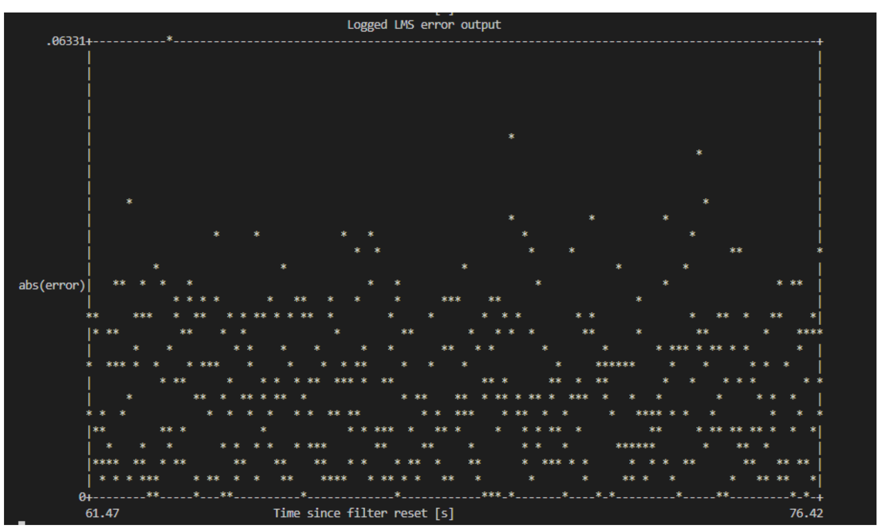

Figure 33. h_{bb} filter error for the broadband disturbance

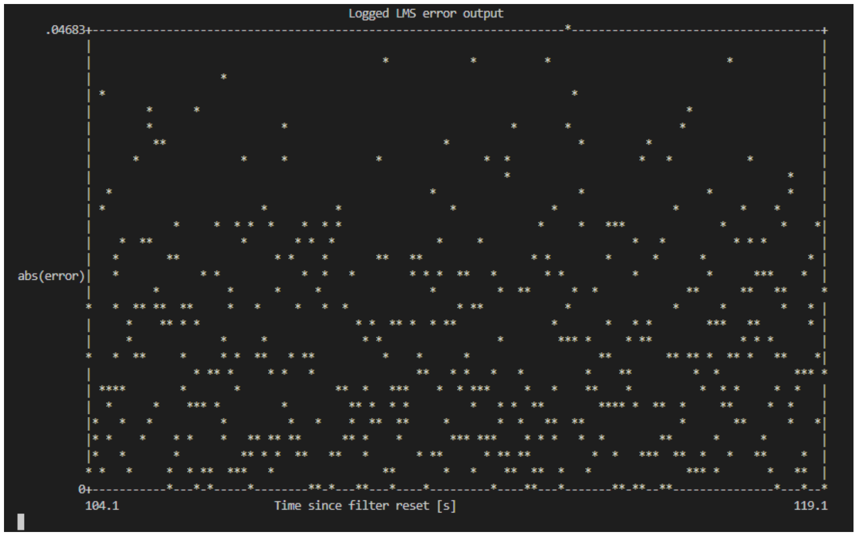

Figure 34. h_{bb} filter error for the sinusoidal disturbance

Comparing the above Figures 31-34 we can clearly see that the h_{sin} filter error for the broadband disturbance is greater h_{bb} filter error for the broadband disturbance approximately by a factor of three which means that the h_{sin} filter does not effectively attenuate the broad-band disturbance compared to the h_{bb} filter which is expected since the h_{sin} filter is trained for the sinusoidal disturbance in the first place. However, when we did the tried the opposite, we can see that the h_{bb} filter error for the sinusoidal disturbance is actually on the same level as the h_{sin} filter error for the sinusoidal disturbance which means that the h_{bb} filter actually attenuates the sinusoidal disturbance effectively.

QUESTION 6) The requested plot can by found in the following Figure

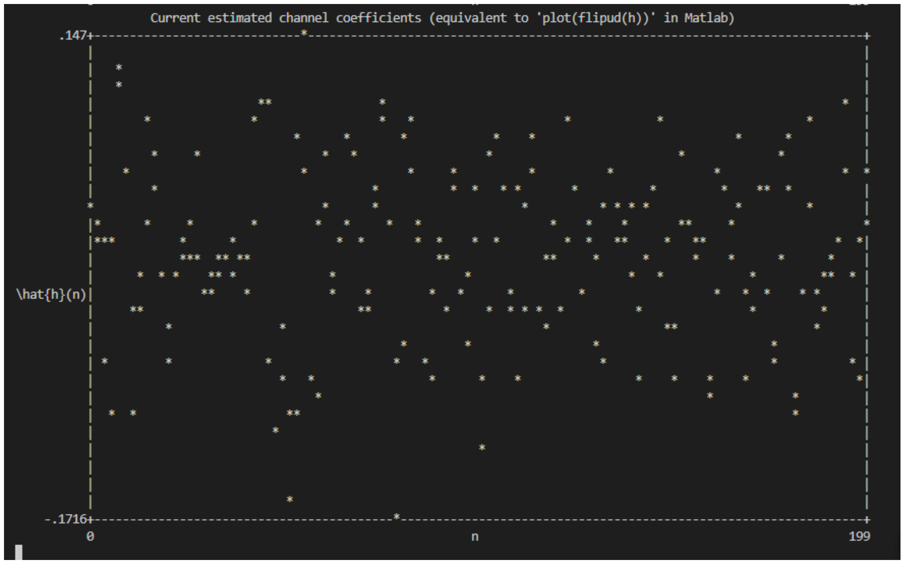

Figure 35. Holding the phone very close to the DSP kit, $h_{bb,sat}$

Comparing Figure 35 and Figure 1 we can clearly see that the $h_{bb,sat}$ filter coefficients are significantly different compared to h_{bb} and the two plots are significantly different as well. The $h_{bb,sat}$ filter coefficients are significantly higher and the filter does not seem to converge in this case. The above happens because we are actually saturating the microphone input with the loud music and so clipping of the music signal occurs. However clipping is not a linear operation, it is actually a non-linear operation since it messes with the ability of the microphone to sample the air pressure. In view of the above it introduces non-linear errors which cannot be compensated from our LMS filter which assumes a linear channel to work optimally.

Analytical section:

QUESTION 1.a) A larger step size μ means that the filter adapts quicker to the channel while a lower step size μ means that the filter adapts slower to the channel. We can confirm same by looking at the relevant plots.

QUESTION 1.b) A large enough step size μ will cause the estimated channel to diverge. If we look at equation (13) of the assignment PDF we will actually see that the step size μ is multiplied to the error signal e(n) when calculating the estimated channel h_hat, so a large enough step size μ could lead the estimated channel to diverge.

QUESTION 1.c) The step size μ does affect the filter accuracy as well. A larger step size μ leads to smaller accuracy of the filter after convergence while a smaller step size μ leads to higher filter accuracy after convergence.

QUESTION 2) For the BB Noise Source we seem to be getting good enough perfomance for different values of length which seems to not be happening the Sinusoidal case. In the first case of the Broadband Noise and since we have Gaussian noise which fluctuates between some certain amplitude levels, there are many solutions that minimise the LMS problem and thus different filter coefficients can give good results for the same level of Gaussian Noise. However this is clearly not the case for the Sinusoidal noise where the solution that minimises the LMS objective is clearly unique.

QUESTION 3) The requested plots can be found in the following Figures 36 – 37

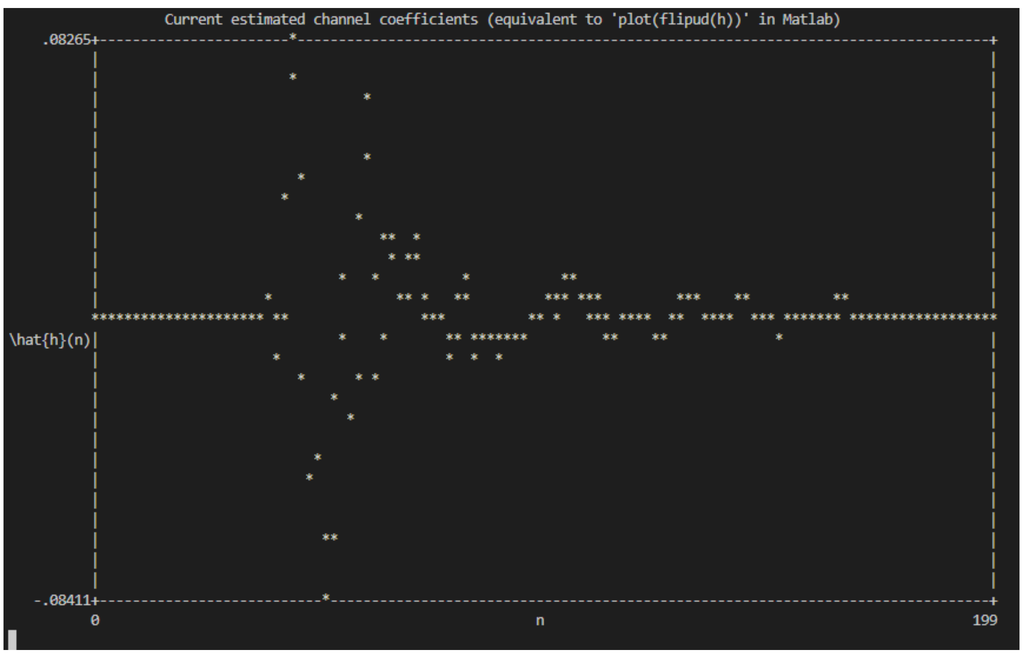

Figure 36. Channel Estimate $h_{sin->bb}$ Channel Coefficients

Figure 37. Channel Estimate $h_{bb->sin}$ Channel Coefficients

If we compare Figures 1 and 2 to Figures 36 and 37 we can clearly see that the above channel estimates look like a combination or a convolution of the channel estimates for h_{bb} and h_{sin}. Figure 36 looks like it was created by Figure 2 as a base and Figure 1 as acting upon Figure 2 and Figure 37 is the opposite. The above plots look the way they look due to the fact that we want to minimise the arguments of the mean-squared error criterion by utilising the LMS algorithm and one of the parameters that we want to minimise is the Y parameter which is the fourrier transform of the y(n) vector which is actually a vector that contains the present and past M-1 input signals y(n). In view of the above the LMS solution will always be affected by the recent previous time signals and thus the resulting coefficients will be affected by the previous values which correspond to the other kind of disturbance. Due to the above, the resulting plots are expected to be convoluted.

QUESTION 5) Considering the LMS objective function, we need to have H – H_hat = 0 when the noise source y[k] = 1 and in this way we will be able to minimise the h_hat_opt function. We can easily prove that for k = (n + ¼) * fs / f0 , where fs = sampling frequency, f0 = 440 Hz and n: integer we have y[k] = 1 and we need to have H – H_hat = 0 at the corresponding frequencies.

APPENDIX:

```
int n;
int i;
    for (n=0; n<block_size; n++)
    {float* y_book = &lms_state[n];
    arm_dot_prod_f32(lms_coeffs,y_book,lms_taps,&xhat);
    e[n] = x[n] - xhat[n];
for (i=0; i<lms_taps; i++) {
    lms_coeffs[i] += 2 * lms_mu * y_book[i] * e[n];
}
    }
```