Chalmers University of Technology

SSY130 - Applied Signal Processing

Project 2: Adaptive Noise Cancellation

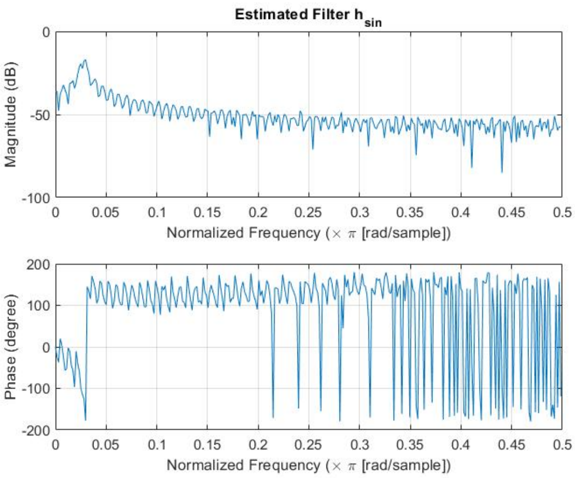
Project Report 2: Group 5

Student names:
Volanka Weerasinghe Arachchige
Akshay Seethanadi Aravind
Mukundh Balabhadra
Yogith Madha

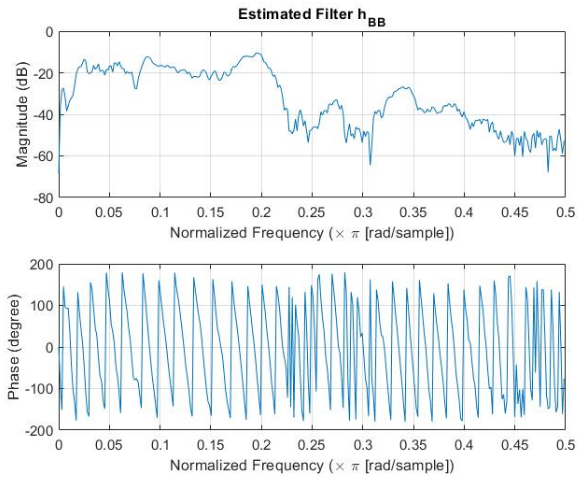
Empirical Section

Question 1

Maximum number of taps achieved with our LMS algorithm - 320.

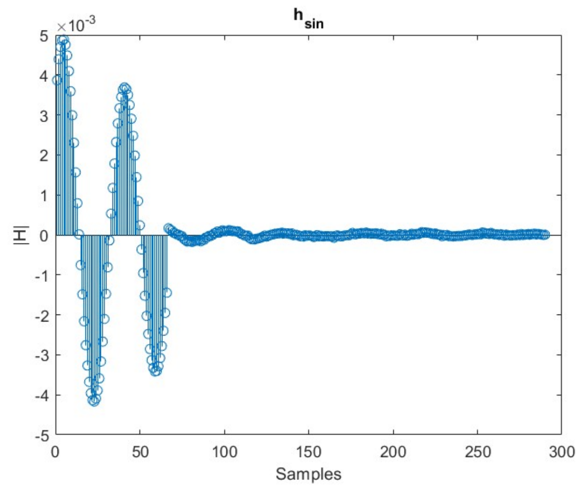


(a) $\hat{H}_{sin}$ Filter

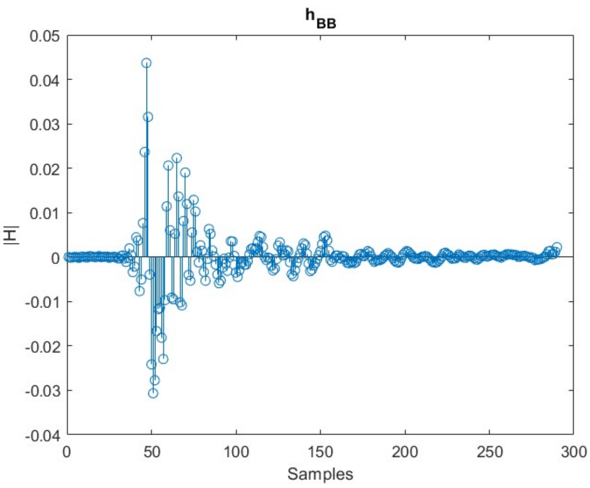


(b) $\hat{H}_{BB}$ Filter

Figure 1: Magnitudes and wrapped frequencies



(a) $\hat{h}_{sin}$ filter



(b) $\hat{h}_{BB}$ filter

Figure 2: Estimated Filter Coefficient

Question 2

(a) Effect due to channel

If we change the input signal or channel without updating the filter values, the error $e(n)$ increases and we start hearing noise. To test our hypothesis, we placed a book between the microphone and the speaker and observed a significant increase in the error values as in figure 3.

This occurs because the real channel $h(n)$ has changed, and our previous estimated channel $\hat{h}(n)$ no longer corresponds to this channel, and thus the DSP will no longer be able to filter the noise properly.

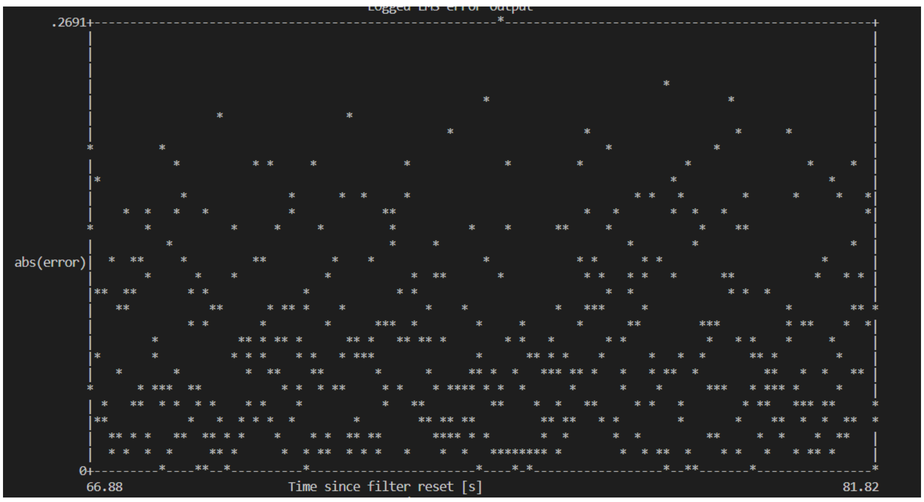
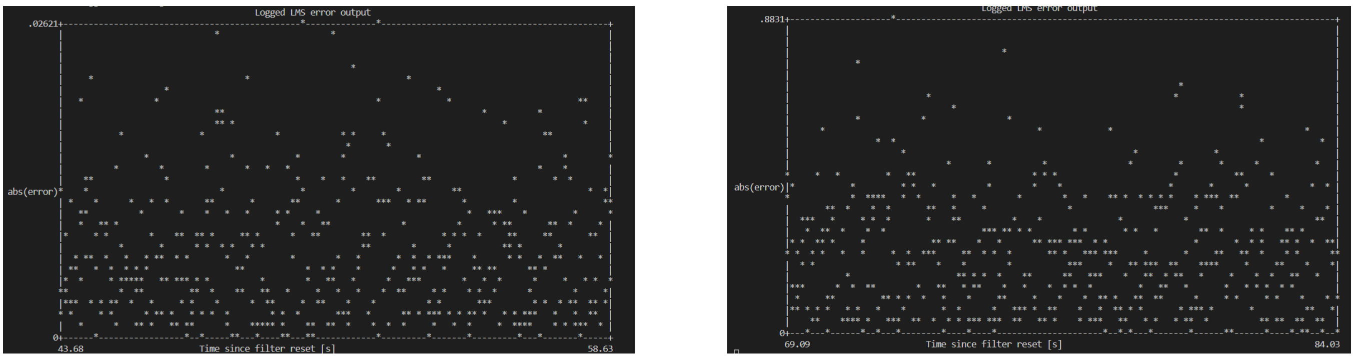


Figure 3: Error when channel is changed for constant $\hat{h}$

(b) Effect due to change in signal amplitude

As the volume of the speakers are increased, the music signal $s(n)$ and the error $e(n)$ are amplified. Clearly, this implies that we have poorer filtering if the filter coefficients are not being updated. Like in case of question a) where the error increases in both cases, the filtering quality also deteriorates in this experiment.
Figure 4 shows the error $e(n)$ before and after increasing the signal amplitude for a constant $\hat{h}$.



(a) Error with the lower signal amplitude

(b) Error with the higher signal amplitude

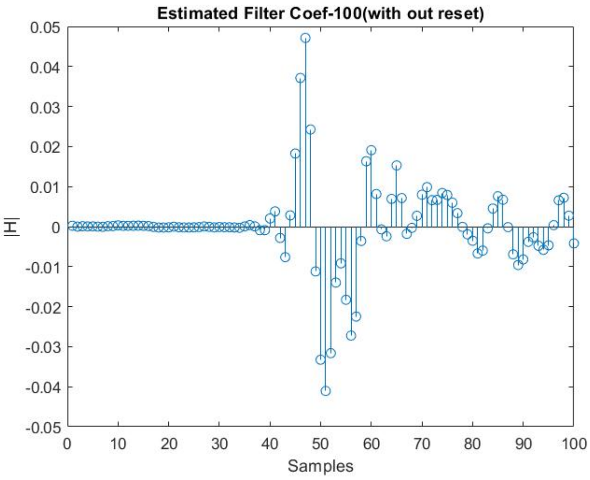
Figure 4: Effect of signal amplitude on error

(c) Effect due to change in source of signal $s(n)$

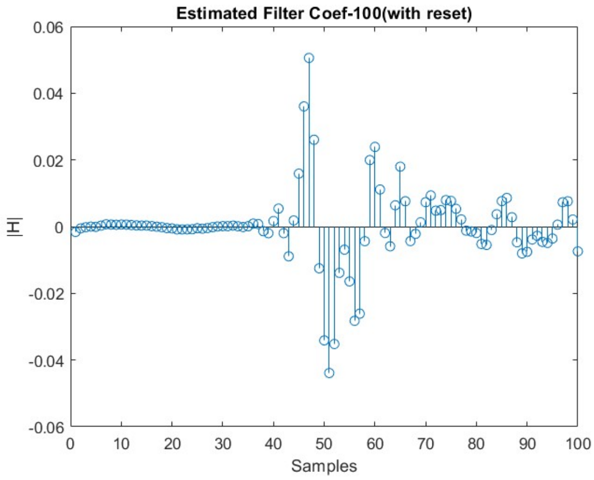
When the source of the music signal $s(n)$ is changed to the cell phone, the filtering works as intended.
When the source of signal $s(n)$ is moved around, the amplitude of the noise doesn't change relative to the moving signal $s(n)$. Therefore, the filter works as intended just that the amplitude of the filtered signal depends on the amplitude of $s(n)$.

Question 3

Resetting or not resetting the filter coefficients after changing the length of the filter has no effect on the noise filtering, since the filter coefficients being updated continuously. Below figures give a clear understanding.



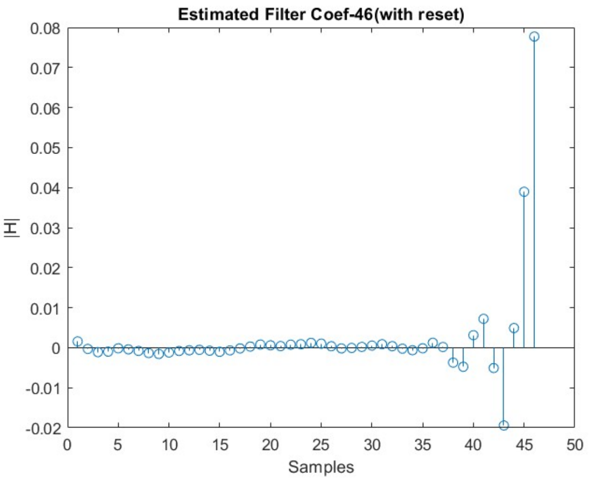
(a) h_{BB}^{100} without reset



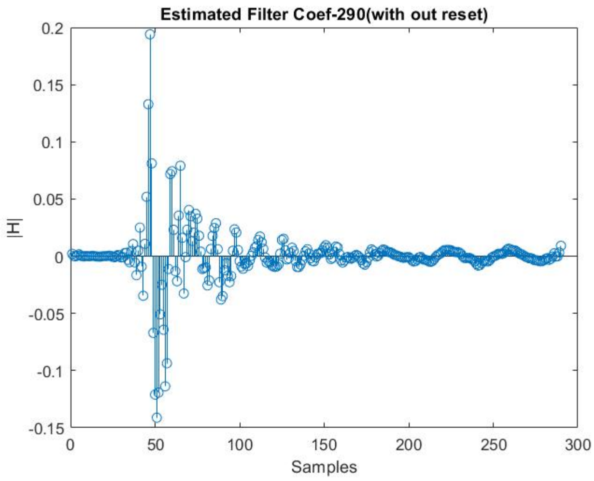
(b) h_{BB}^{100} after reset

Figure 5: h estimated

The filter does not work equally well for all filter lengths, for $\mu = 1 * e^{-3}$ we found good performance with 300 and 100 coefficients, a decent performance with 50, and very poor performance with less than 40 coefficients. Therefore, the critical length for poor performance is around 40 coefficients.



(a) h_{BB}^{46}



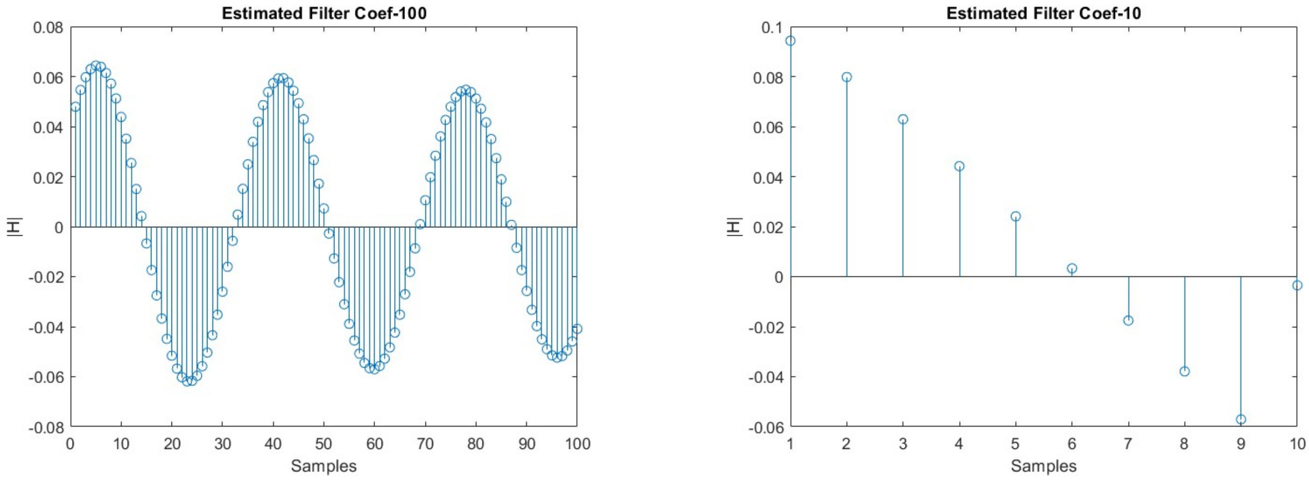
(b) h_{BB}^{290}

Figure 6: h estimated

With reference to the figure 6a, any number of coefficients less than 40 will result in poor filtering, due to the fact that in the LMS algorithm $h(n)$ depends on $h(n-1)$ and they are are close to zero during the time required for learning as seen in figure 2b.

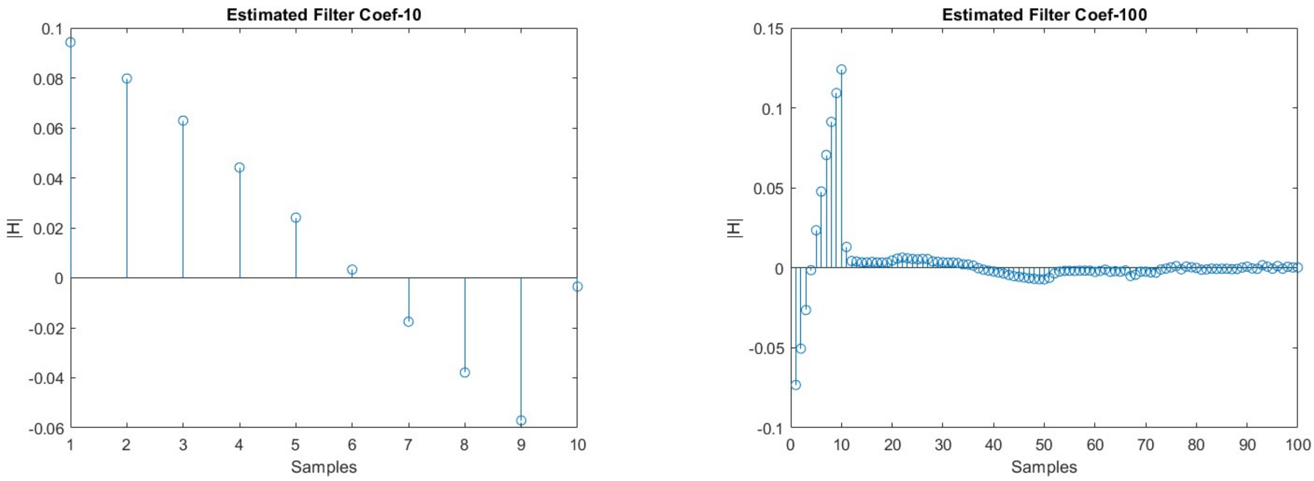
Question 4

When h_{sin} is updating and the filter size reduced from 100 to 10 without resetting, a reduction in the magnitude of the filter coefficients was observed as shown in Figure 7.
When the filter length was increased from 10 to 100 while the filter kept updating, although the number of coefficients increased a significant magnitude change in the added coefficients was not observed as shown in Figure 8.
The filter worked equally well for both length 10 and 100 as we could not observe a significant change in the error.



(a) $\hat{h}_{sin}^{100}$ *without reset* *(b)* $\hat{h}_{sin}^{10}$ *without reset*

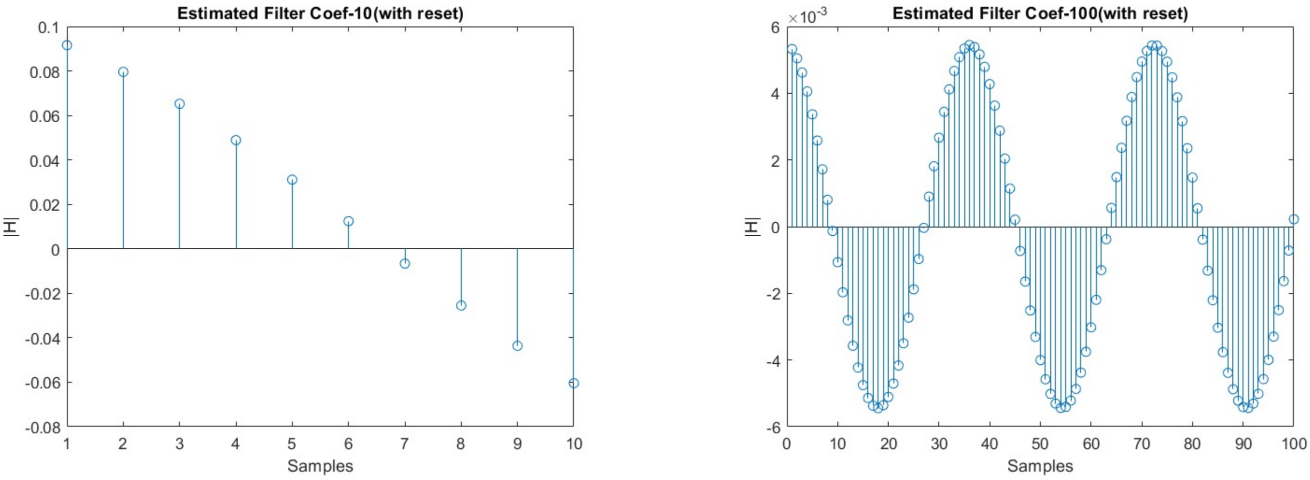
Figure 7: $\hat{h}_{sin}$ *Reducing length from 100 to 10 without reset*



(a) $\hat{h}_{sin}^{10}$*without reset* *(b)* $\hat{h}_{sin}^{100}$ *without reset*

Figure 8: $\hat{h}_{sin}$ *Increasing length from 10 to 100 without reset*

When $\hat{h}_{sin}$ is reset after changing the filter length from 100 to 10, similar result was observed as in the without reset case in Figure 7.
When $\hat{h}_{sin}$ is reset after changing the filter length from 10 to 100, and the result is shown in Figure 9. Significant change in the newly added coefficients (from 10 to 100) can be observed compared to the without reset case in the Figure 8.



(a) $\hat{h}_{sin}^{100}$ with reset *(b) $\hat{h}_{sin}^{10}$ with reset*

Figure 9: $\hat{h}_{sin}$ Increasing length from 10 to 100 with reset

Question 5

On the left speaker it was observed that the $\hat{h}_{BB}$ is able to cancel the sinusoidal noise, but $\hat{h}_{sin}$ is unable to cancel the broadband noise properly.

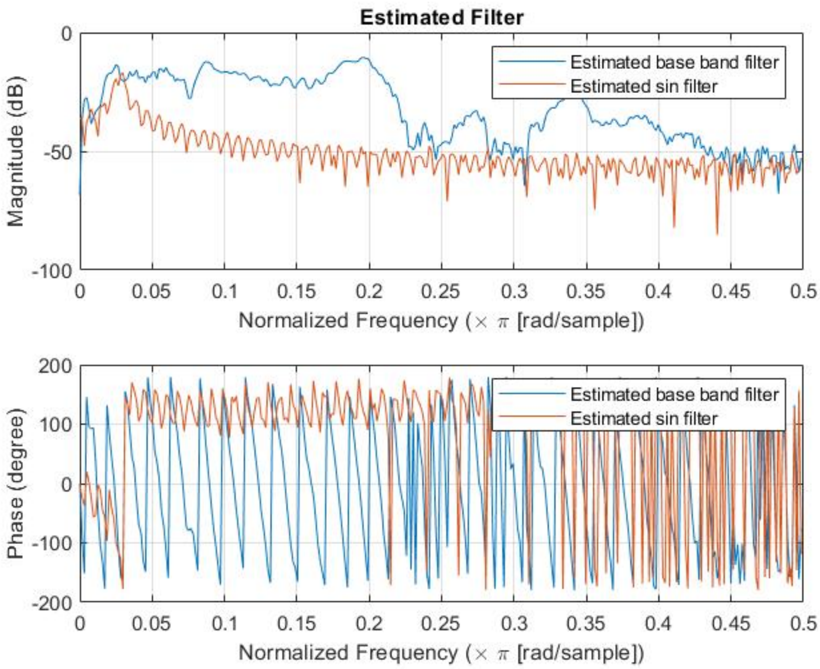


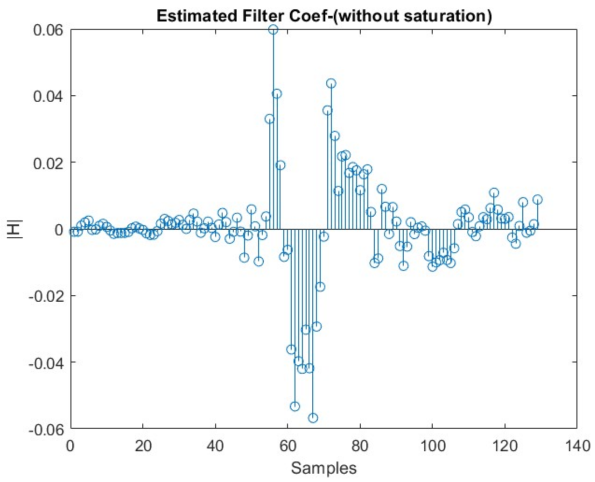
Figure 10: Channel $H_1(k)$ response

The sinusoidal is at frequency 440 Hz and it can be seen in the Figure 10 that the frequency is also included in the $\hat{h}_{BB}$. As a result the $\hat{h}_{BB}$ is able to properly cancel out the sinusoidal noise. In the same way, $\hat{h}_{sin}$ is only properly cancelling the noise frequency around 440 Hz and remaining noise frequencies of the broadband are passed without cancelling.

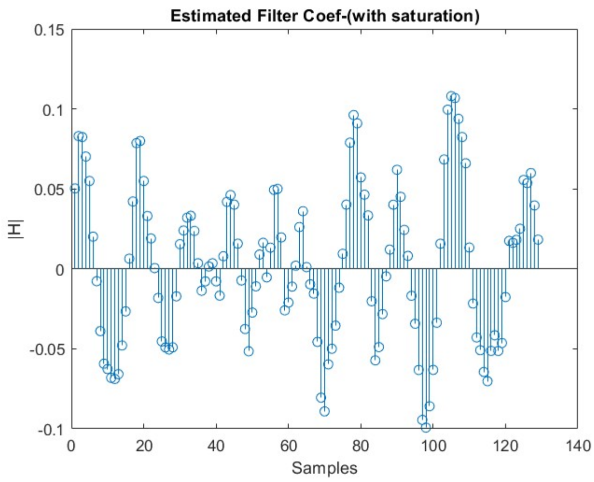
Question 6

With μ set to 10^{-2}, noise set to broadband and loud music was played very close to the mic, the resulting filter is $h_{BB,sat}$ and the filter in unsaturated conditions is h_{BB} are shown in Figure 11.

Saturation creates nonlinear distortions, because LMS adaptive filters can only cancel the linear portions of the signal, the non-linear portions cannot be removed. The way an adaptive algorithm responds to these disruptions is similar to how these algorithms respond to doubletalk situations. In other words, the faster an algorithm converges, the more nonlinearities affect it. Because a whole block of data is degraded when the system diverges, block processing algorithms perform worse in terms of convergence during nonlinearities.



(a) h_{BB}



(b) $h_{BB,sat}$

Figure 11: h_{BB} and $h_{BB,sat}$

Analytical Section

Question 1

(a)

The optimal estimate of the filter $\hat{h}_{opt}$ for a stationary application is achieved by the mean squared error criterion:

$$\hat{h} = arg \min_{h} E[e(n)^2] \tag{1}$$

An approximate solution to the criterion equation 1 is obtained by the LMS algorithm in equation 2 where as μ is the step-size.

$$\hat{h}(n) = \hat{h}(n-1) + 2\mu y(n)e(n) \tag{2}$$

The filter error at sample index n is given by equation 3

$$\tilde{h}(n) = \hat{h}_{opt} - \hat{h}(n) \tag{3}$$

The size of μ needs to make $\|\tilde{h}(n)\| \to 0$ as $n \to \infty$

$$\tilde{h}(n) \approx (I - 2\mu\Phi_{yy})^n \tilde{h}(n) \tag{4}$$

Where as Φ_{yy} is the auto-correlation matrix of the signal.
The step-size μ decides how quickly the $\hat{h}$ converges to $\hat{h}_{opt}$ hence it decides how quickly it adapts to the channel.

(b)

As we increase the size of the μ, the convergence rate increases, but the stability characteristics decrease, making the system more likely to diverge rather than converge to the optimal solution.

(c)

As μ value increases the mean square error value between the converged $\hat{h}$ and the $\hat{h}_{opt}$ increases. Higher the MSE value, the accuracy of $\hat{h}$ resembling the channel degrades, hence the filter performance degrades.
Step-size μ decides not only the convergence rate, but also how far away the converged $\hat{h}$ stays from $\hat{h}_{opt}$ where as high μ results high residual variance.

Question 2

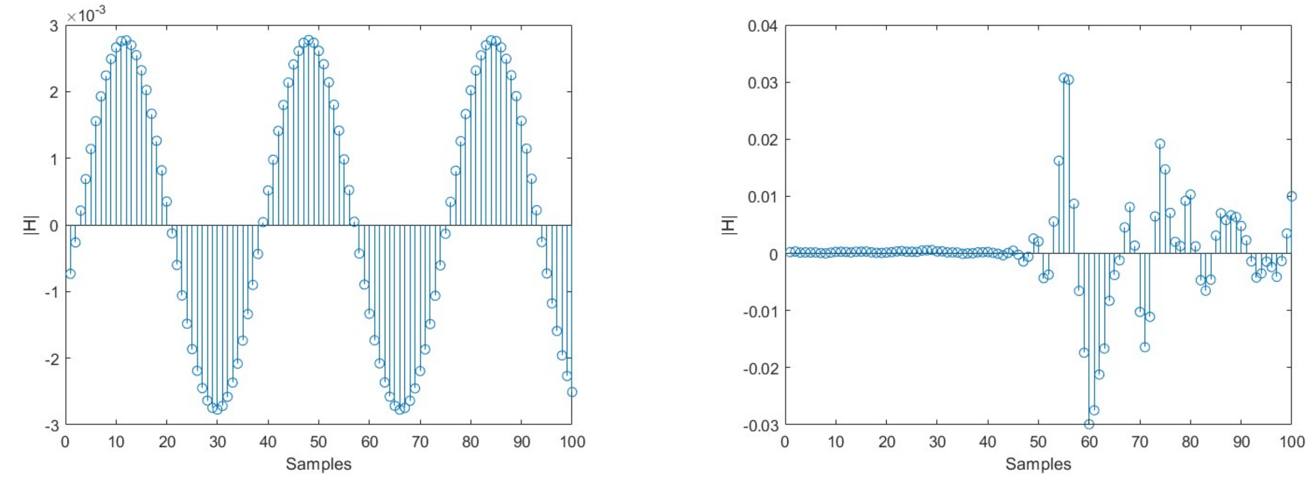
Recalling the LMS algorithm,

$$\hat{h}(n) = \hat{h}(n-1) + 2\mu y(n)\underbrace{\left(x(n) - \hat{h}^T(n-1)y(n)\right)}_{e(n)} \tag{5}$$

we can clearly see that $\hat{h}(n)$ is dependent on $y(n)$. $y(n)$ is the noise signal we receive.

Since the noise signals are a sinusoidal function of constant frequency and a broad-band function which uses random frequency wave, while testing for h_{sin}, we are expected to receive $\hat{h}_{sin}$ in terms of sinusoidal functions depending on the taps of the filter. Whereas, h_{BB} is a upsampled Gaussian signal, hence $\hat{h}_{BB}$ depends on this random sequence. This is the reason for the drastically different behaviors for the different sources of noise.

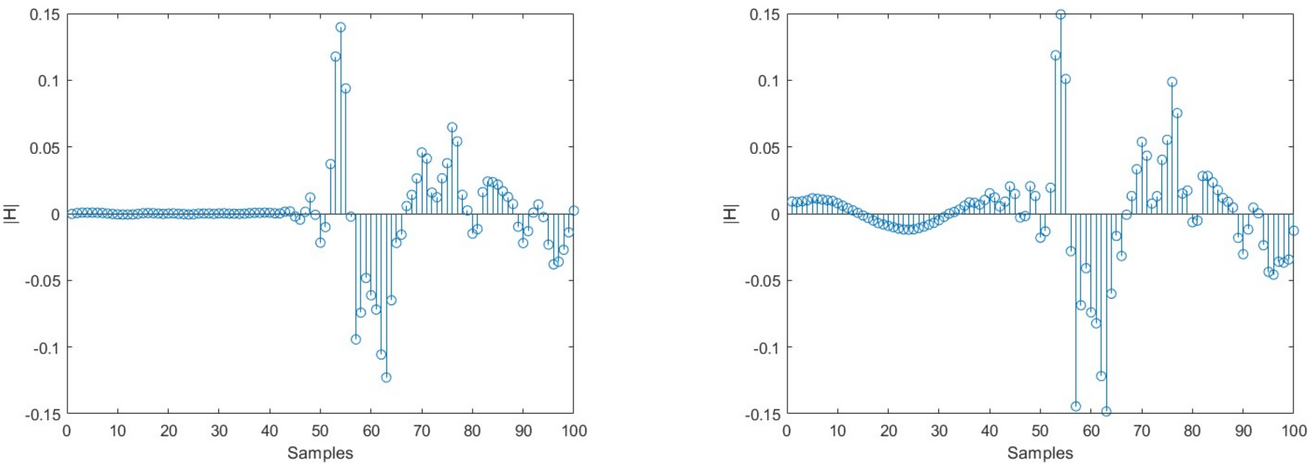
Question 3



(a) h_{sin} *(b) $h_{sin\to BB}$*

Figure 12: h_{sin} and $h_{sin\to BB}$

When the trained LMS filter for the sinusoidal disturbance h_{sin} is exposed to the broadband noise, the filter changed significantly as shown in 12. The reason for this significant change is that the h_{sin} was adapted to the sinusoidal noise which was 440 Hz but the broadband frequency range covers more range in the frequency spectrum. h_{sin} cannot handle the broadband noise hence $h_{sin\to BB}$ is significantly different to the h_{sin}.



(a) h_{BB} *(b) $h_{BB\to sin}$*

Figure 13: h_{BB} and $h_{BB\to sin}$

When the fully trained LMS filter for the broadband noise h_{BB} is exposed to the sinusoidal noise, significant change in the filter cannot be observed as shown in 12. The reason for this is that the h_{BB} was adapted to the broadband noise where it also includes 440 Hz sinusoidal noise. As the broadband frequency range covers the sinusoidal noise, h_{BB} can handle the sinusoidal noise hence $h_{BB\to sin}$ is similar to the h_{BB}.

Question 4

When $f_o = 440Hz$,
$H_{sin}(f_o) = 0.0891\angle 144°$
$H_{BB}(f_o) = 0.0978\angle 155°$

By Referring to the figure 10, we understand that the filter will try to match the amplitude and phase of the real channel H for **440Hz** frequency. So, we expect the amplitudes and phases of $H_{sin}(f_o)$ and $H_{BB}(f_o)$ to be similar. On other hand Figure 10 for the rest of other frequency we observed that sin filter can cancel only certain frequency of noise, where as broad band filter can filter larger frequency of noise.

Question 5

The optimal filter is defined as

$$h_{opt} = arg\ \underset{h}{min} \frac{1}{2\pi} \int_0^{2\pi} |H(\omega) - \hat{H}(\omega)|^2 S_{yy}(\omega) d\omega \tag{6}$$

where S_{yy} is power spectral density of the input signal $y(n)$.
Since $y(n)$ is a sinusoidal disturbance with frequency $\omega_0 = 2\pi f_0$. The power spectral density of the disturbance will correspond to two delta functions at location ω_0 and $-\omega_0$. To fulfill the LMS objective, The filter $\hat{H}$ only needs to match the amplitude and phase precisely at frequency ω_0 [1].

Since $y(n) = sin(\omega_0 n) = \frac{1}{2j}(e^{j\omega_0 n} - e^{-j\omega_0 n})$ is a complex number, the DTFT of filter $h(n) = [h_0, h_1, ..., h_N]$ given as

$$H(\omega) = \sum_{n=0}^{M-1} h(n) e^{-j\omega \Delta t n} \tag{7}$$

should also be a complex valued function. The minimum value matching this criteria is with N=2. Therefore, the minimum number of coefficients required are 2.

Question 6

The LMS algorithm is defined as,

$$\hat{h}(n) = \hat{h}(n-1) + 2\mu y(n) \underbrace{(x(n) - \hat{h}^T(n-1)y(n))}_{e(n)} \tag{8}$$

As we know that $y(n)$ corresponds to a sine function of frequency 'f_0', the term $2\mu e(n)$ is therefore a multiple of y(n). Thus $\hat{h}(n)$ is a sum of sinusoidal signal with varying magnitude. As we can see from Question 4, figure 8 $\hat{h}(n)$ is a sinusoidal wave with varying magnitude but with the same constant frequency.

[1] T. McKelvey *SSY130 - Applied Signal Processing Lectures notes*

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+n);
    e[n] = x[n] - xhat[n];

    for(i=0;i<lms_taps;i++){
        lms_coeffs[i]+= 2 * lms_mu * y_book[i] * e[n];
    }
}
```

Figure 14: LMS algorithm