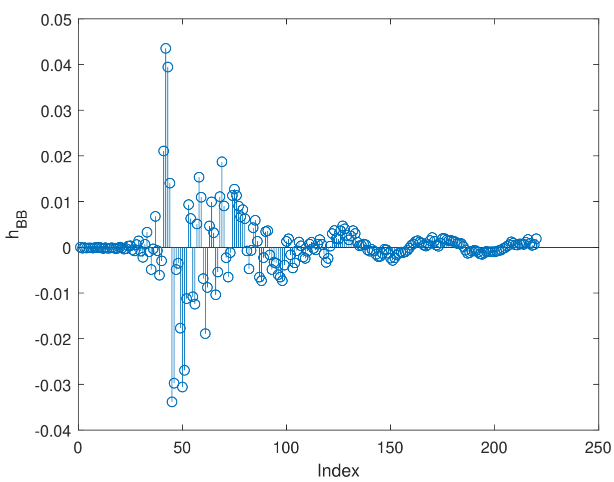
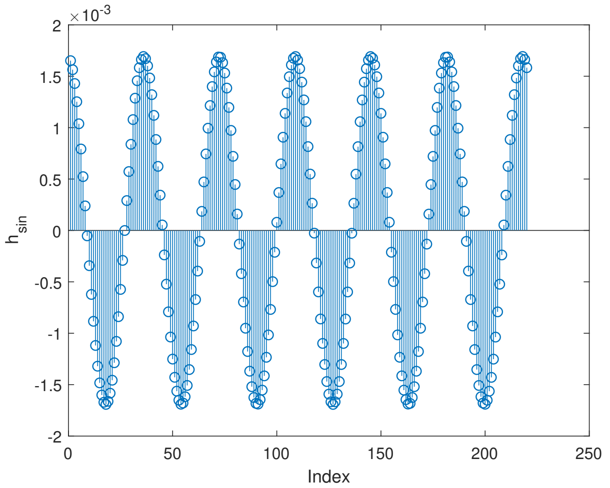
Empirical section:

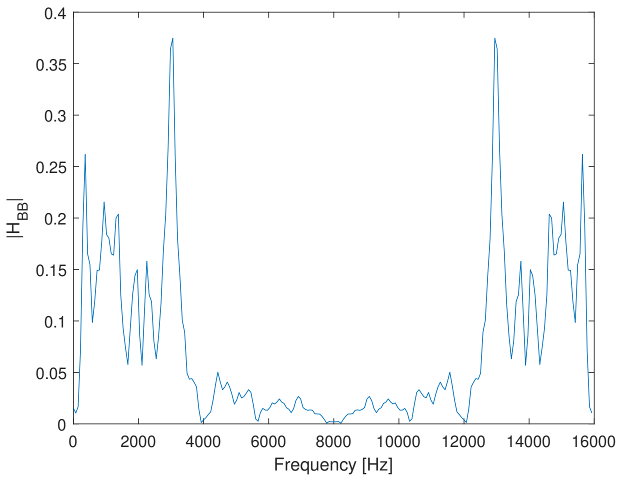
1.



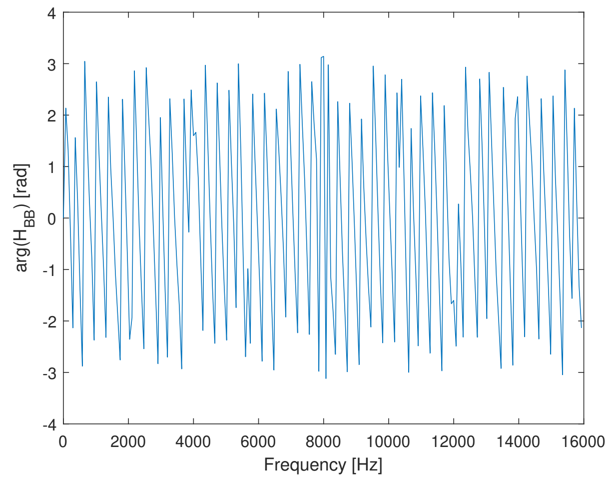
(a) impulse response:h_{BB}



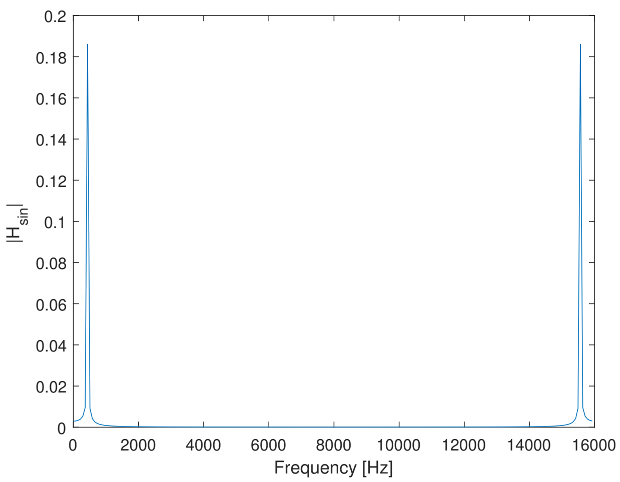
(b) impulse response:h_{sin}



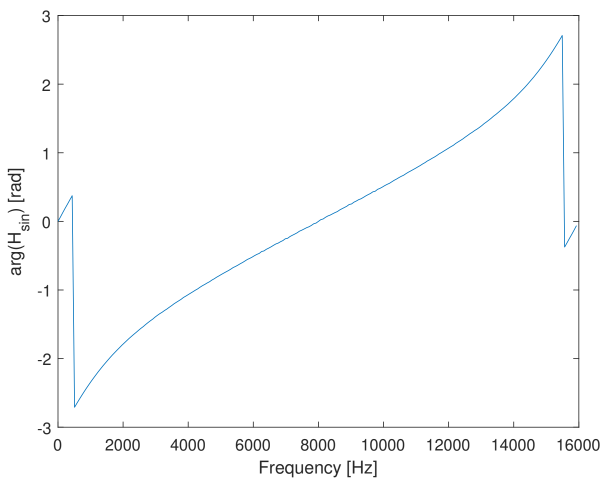
(c) amplitude response:$|H_{BB}|$



(d) phase response:$arg(H_{BB})$



(e) amplitude response:$|H_{sin}|$



(f) phase response:$arg(H_{sin})$

Figure 1

2.

a)

Without updating the coefficients of the estimated channel, the output error will increase, because the new channel does not have the same response as before. That is, $|\mathbf{h} - \hat{\mathbf{h}}|$ increases and so also $e(n)$. This can be seen in figure 2 and 3 that the y-axis amplitude increased after changing the channel without updating the filter.

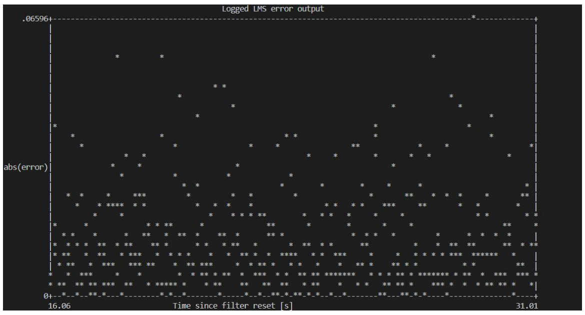


Figure 2: Before changing the channel

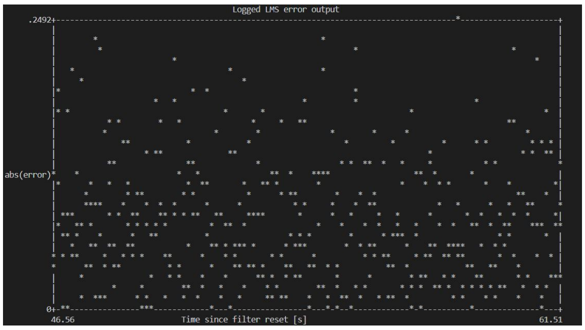


Figure 3: After changing the channel

b)

By keeping the filter coefficients constant without updating them, increasing/decreasing the volume will increase the output error. This result agrees with the conclusion in question 2a because changing the volume can be considered as the change in the channel and since the equalizer taps are not updated, the error signal will increase.

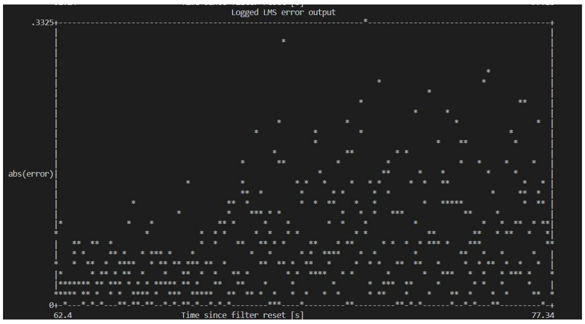


Figure 4: Decreasing the volume

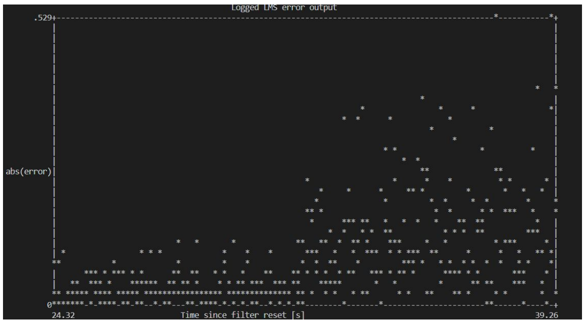


Figure 5: Increasing the volume

c)

In this case, since the channel for the noise source is still constant, the LMS algorithm can suppress the noise and the output error function will be equal to the music from the cell phone. Moving the cell phone around will change the channel for the music but still the channel is almost the same for the noise. Therefore, the noise can be removed.

3.

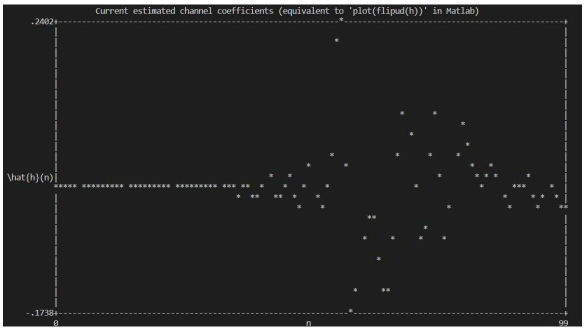
Based on your results, in the steady state, the results of both cases for resetting the filter and not resetting the filter, are the same, and we see the same scale of error in both cases.

The quality of noise cancellation is not the same for different length of the filter. Decreasing the number of taps will increase the error and noise cancellation will not be perfect.

Based on question 1, the minimum length of equalizer should be somewhere between 100 to 150, to compensate for the effect of the channel. This is the region that most of the channel response energy exist.

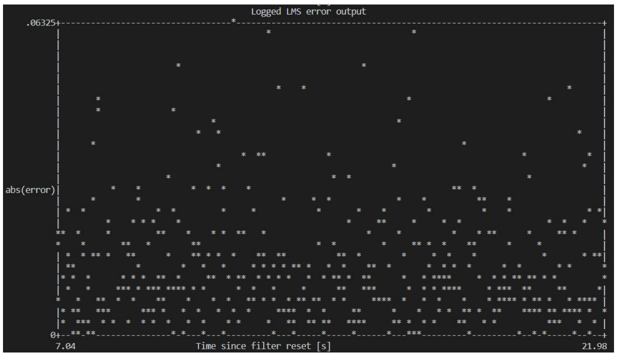
When we reduce the number of taps to below 65 taps, the quality of noise cancellation drops significantly. This can be related to delay in the transmission and reception of the sound signal.

(a) 220 taps

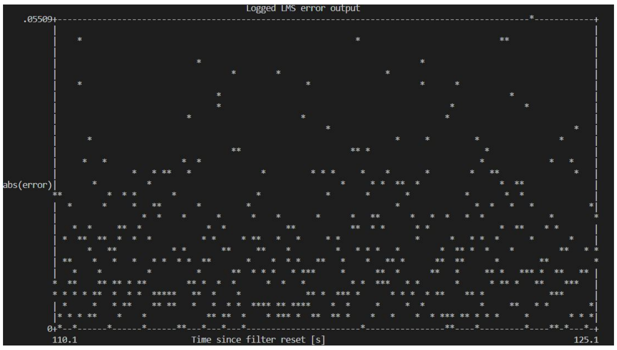


(b) 100 taps

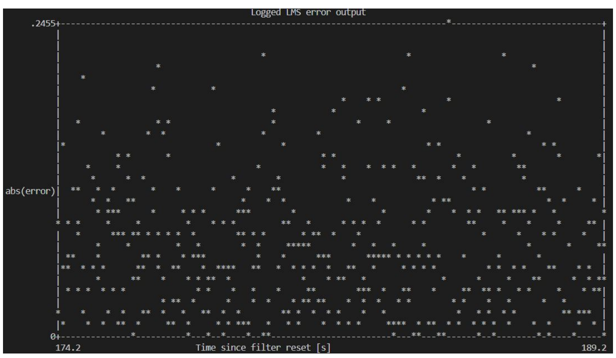
Figure 6: Filter coefficients with different number of taps



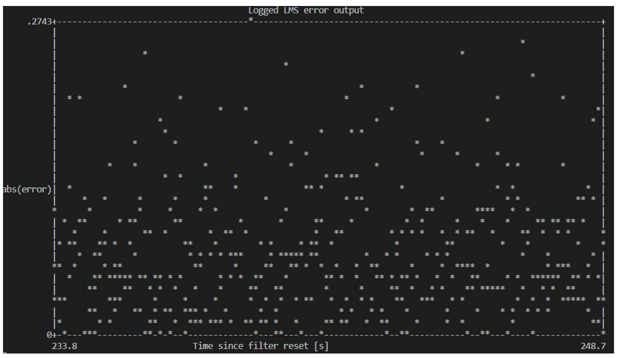
(a) 220 taps



(b) 100 taps



(c) 50 taps

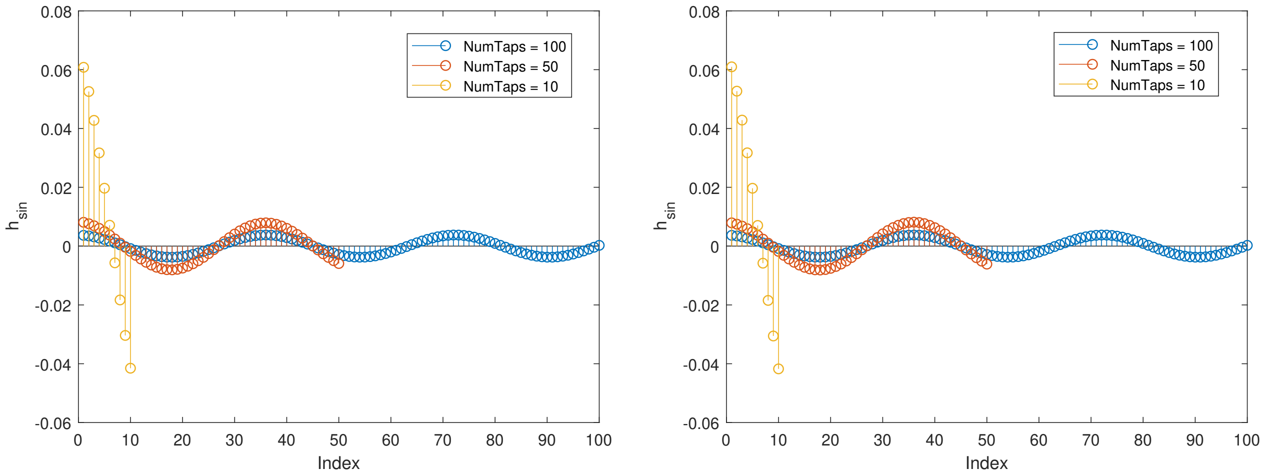


(d) 25 taps

Figure 7: Error signal

4.

a) If we change the coefficients number from 100 to 10, the sine shape of h_{sin} keeps unchanged and h_{sin} is enlarged by a given scalar. Resetting operation will not affect the converged filter h_{sin} coefficient and also the filter performance. We observe similar error value for these lengths, so filters with different length have equal performance.

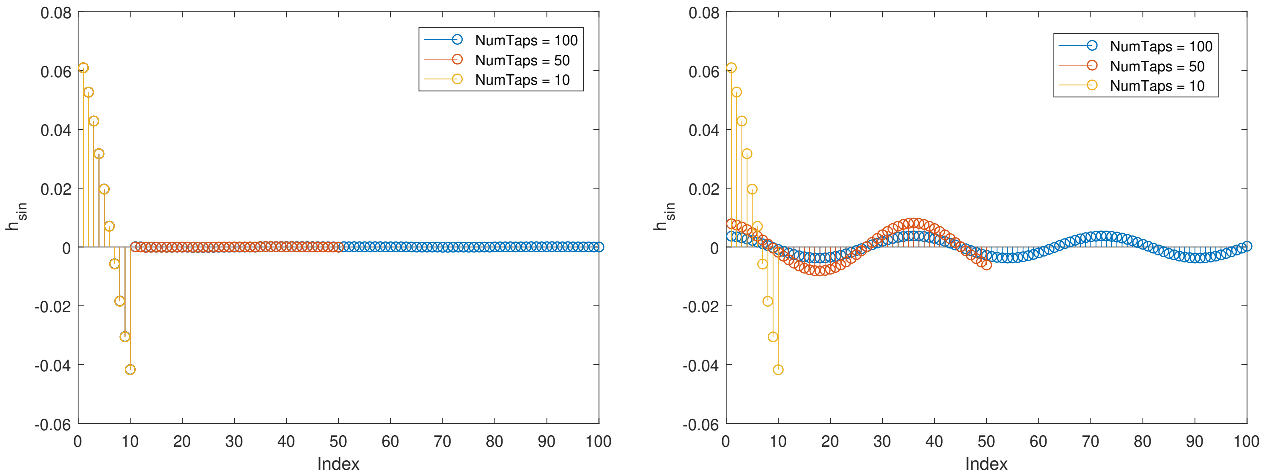


(a) w/o resetting coefficients

(b) Resetting coefficients

Figure 8: Plot of h_{sin} when changing from 100 to 10 elements

b) If we change from 10 to 100 elements, the filter will have zero padding operation and these padded coefficients keep the same even after training (e.g. Always zero). However, h_{sin} changes back to sine wave shape if we have done a reset operation after increasing the lengths. In this case, all these filters still have similar error output after being well-trained. It denotes that zero-padding and sine-shape filters have the same performance.



(a) w/o resetting coefficients

(b) Resetting coefficients

Figure 9: Plot of h_{sin} when changing from 10 to 100 elements

5.

a) When trained and fixed LMS filter h_{BB} is used for sinusoidal noise, the error output is unchanged, which denotes that h_{BB} works well for sine disturbance and have similar performance as h_{sin} filter. On the other hand, as shown in Figure 8, h_{sin} the filter cannot fully attenuate broad-band noise and have a larger error output.

b) The reason is that, based on $LMS\ objective$(12), we should design filter h to eliminate $Y(H - \hat{H})$ such that error is minimized. h_{BB} is trained too close to the real channel response h at frequency $[0, 2000]Hz$, where the majority of the broad-band noise power is located. However, since sinusoidal noise has single frequency term $f_0 = 440Hz$, h_{sin} is only required to be close to the real channel response at f_0. f_0 is included in frequency range $[0, 2000]Hz$, so h_{BB} works well for sine noise. Nevertheless, h_{sin} does not maintain the correct real channel response in the full spectrum of broad-band disturbance. So h_{sin} cannot fully attenuate the Broad-band noise.

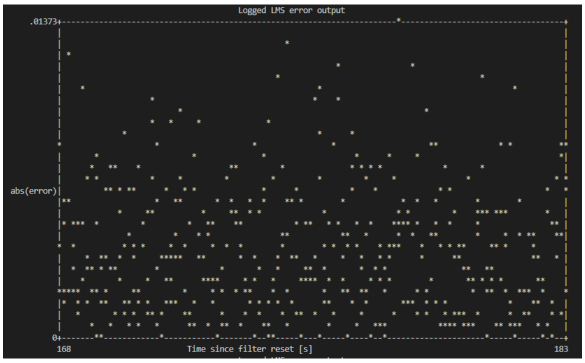


Figure 10: h_{BB} used for sinusoidal noise

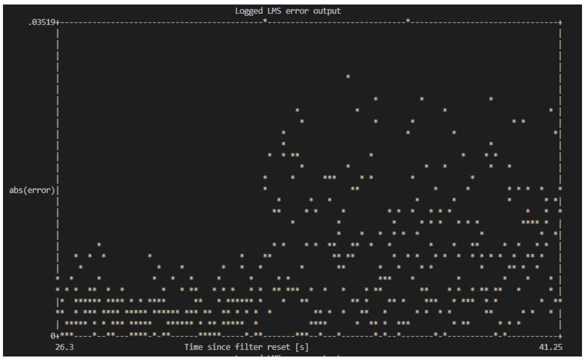


Figure 11: h_{sin} used for broad-band noise

6.

$h_{bb,sat}$ is different from h_{bb} as can be seen in figure 12 but there are some similarities as well especially on the location of high oscillation of the channel response. But, the other filter coefficients have changed. This is because LMS is trying to equalize a non-linear channel and the saturation behaviour with a linear filter. Therefore, the coefficients which are supposed to be zeros will change to compensate the non-linear effects as much as possible which might not be so effective because the filter will still be linear. This shows why LMS basically does not compensate for non-linearity, except we have a further modification of the system.

Figure 12: Filter coefficients in the saturated mode ($h_{bb,sat}$)

Analytical section:

1.

a) A larger step size makes the filter to take shorter time to adapt to the channel generally, so a proper step size should be chosen to ensure reasonable convergence speed. A small value will take a longer time to converge while a large value will lead to a faster convergence. A larger step size does however imply a larger convergence zone. Analysing the LMS-convergence rigorously it is clear that a choice of μ is completely given by the auto-correlation of the measured signal y for the regular case. The eigenvalues of the auto-correlation matrix yields information on the optimal choice of μ.
b) Given the problem formulation and solution method it is required for the step length μ to be less than the inverse of the greatest eigenvalue of the auto-correlation matrix. If violated, convergence of the filter is not guaranteed. One can see that for each iteration the step is proportional to the error signal as well as μ and if the error signal increases from taking too long steps the process stimulates itself and divergence is inevitable.Furthermore,if the selected value is too large the LMS algorithm will still diverge.
c) Yes, the choice of the step size affects how well the filter works. As stated previously, the step size determines the size of the convergence zone, that is, with what residual error the solution oscillates around the optimal solution. Here, a smaller step size ensures a smaller convergence zone (residual error) which is generally better (yielding smaller error). Clearly,a large step length results into a larger residual variance, and vice-versa.

2.

a) When we decreased the filter lengths from 100 to 25 in the case of the broad-band noise, the error output increased. However, in the case of the sinusoidal noise, changing the filter lengths did not affect filter performance. The reason is that the broad band noise compensating filter requires many more coefficients to accurately minimise the error whilst an optimal filter may be found already at 10 coefficients for the sinusoidal noise. Based on Empirical section, question 3, the minimum length of equalizer should be somewhere between 100 to 150 to compensate for the channel effect for the broad-band noise.

b) There is no difference whether we reset coefficient in the broad-band noise case, when the length changes between 10 and 100 elements, the filter still assumes the same shape. However, as shown in figure 8, zero-padding and sine-shape filter coefficients are generated depending on whether we are resetting or not. Based on the hint, these filters have equal performance since the solution that minimize the LMS objective (12) is not unique. i.e The error is already minimised for the 10 coefficients and so no update is made since $e(n) = 0$. The reason why resetting gives another shape is because this solution is closer to the initial, also see the last two previous questions.

3.

The obtained channel for both cases are similar to h_{BB}.

In the first case, when the filter is optimized for sinusoidal disturbance, changing the noise source will change the equalization filter because the LMS algorithm tries to compensate for other frequency components as well. That is why in the end, the equalizer filter looks like h_{BB}.

In the second scenario, when the filter is optimized for the broad-band disturbance, changing the noise source will not affect the filter taps because in the broad-band noise that special frequency has already been compensated and the error is zero. Therefore, the filter taps would not change in this case and they will remain as they were.

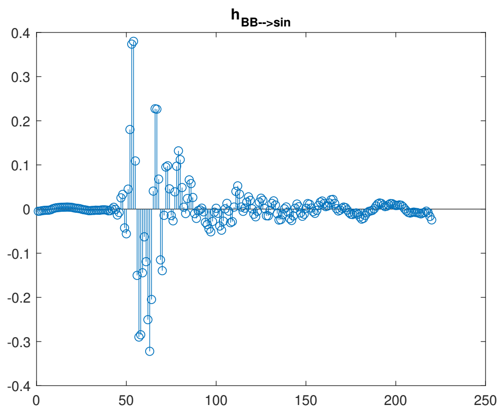


Figure 13: $h_{BB-->sin}$

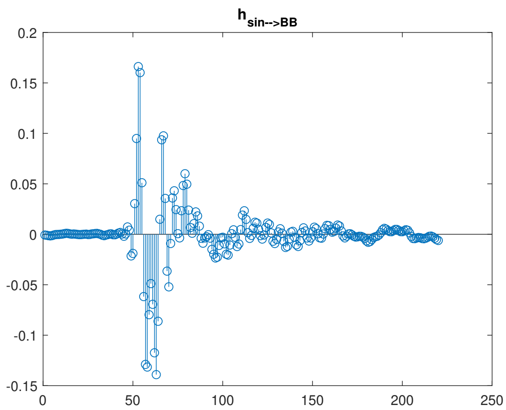


Figure 14: $h_{sin-->BB}$

4.

The magnitude and phase of $H_{sin}(f_0)$ is 0.19 and 0.37 rad, while the magnitude and phase of $H_{BB}(f_0)$ is 0.17 and 0.35 rad. They should have similar value, but we notice that a small difference in value exists between them. We believe $H_{sin}(f_0)$ have more accuracy at f_0 because, based on LMS objective 12, H_{sin} is only required to minimize error in this single frequency. H_{BB} needs to eliminate $Y(H-\hat{H})$ in the range $[0, 2000]Hz$, so it does not have such accurate estimation at frequency f_0. Since Y_{BB} and Y_{sin} only has common frequency tone f_0, H_{BB} should have similar estimation at frequency f_0 compared to H_{sin}. However, they have a totally different estimation within the other frequency range. The reason for this lies in the spectrum of the noise source $Y(\omega)$ for which you minimise the variance. Whatever the initial value h has on frequencies outside of $Y(\omega)$ will not matter for that is not part of the minimisation problem (eq. 12). Thus the shape of $\hat{H}$ at frequencies outside of $Y(\omega)$ will remain, hence the difference.

5.

Since the input is a single frequency ($f_0 = 440Hz$) noise, the response of the linear channel in that specific frequency is important to be compensated for. Let's assume that the channel response is $H(\omega_0) = Ae^{-j\phi}$. Therefore, we need an equalization filter which becomes equal to this transfer function at this frequency. Since, the frequency response of the channel is a complex number with an amplitude and phase, the equalization filter should have the same value in frequency domain. With only one filter coefficient, it will be impossible to make the desired complex response in the frequency domain but with two (2) coefficients we can generate a complex number with a certain amplitude and phase as explained below

$$\begin{aligned}\hat{h}[n] =& \hat{h}[0]\delta[n] + \hat{h}[1]\delta[n-1]\\ \hat{H}(\omega) =& \hat{h}[0] + \hat{h}[1]e^{-j\omega\Delta t}\end{aligned}$$

Then, $H(\omega_0) = \hat{H}(\omega_0)$.

$$Acos(\phi) - jAsin(\phi) = (\hat{h}[0] + \hat{h}[1]cos(\omega_0\Delta t)) - j(\hat{h}[1]sin(\omega_0\Delta t))$$

Solving the above equations will result in

$$\begin{aligned}\hat{h}[1] =& \frac{Asin(\phi)}{sin(\omega_0\Delta t)}\\ \hat{h}[0] =& A(cos(\phi) - sin(\phi)cot(\omega_0\Delta t))\end{aligned}$$

So with the number of coefficients, N = 2 , at least, we can cancel a sinusoidal disturbance. Therefore, these two coefficients, at least, are enough to compensate for the channel response.

6.

Properly using LMS to train the FIR filter for sinusoidal disturbances the plot of the coefficients, with long enough (e.g.Longer then the absolute minimum) filter coefficients, will look like a sine. Referring to filter update step $\hat{\boldsymbol{h}}(n) = \hat{\boldsymbol{h}}(n-1) + 2\mu \cdot \boldsymbol{y}(n) \cdot (x(n) - \hat{\boldsymbol{h}}^T(n-1)\boldsymbol{y}(n))$. We notice that the error term $e(n) = x(n) - \hat{\boldsymbol{h}}^T(n-1)\boldsymbol{y}(n)$ will always be a scalar. For the first case of updating the filter coefficients, $\hat{\boldsymbol{h}}(1) = 2\mu \cdot e(1) \cdot \hat{\boldsymbol{y}}\boldsymbol{(1)}$. Since original sine noise is expressed as $y(n) = sin(2\pi n \frac{f_0}{f_s})$, the vector $\hat{y}(n)$ is given by a sine shape. Then, the updated filter coefficients $\hat{\boldsymbol{h}}(1)$ have a sine-like plot. For the second case of updating the filter coefficients, $\hat{\boldsymbol{h}}(2) = 2\mu \cdot e(2) \cdot \hat{\boldsymbol{y}}\boldsymbol{(2)} + \hat{\boldsymbol{h}}\boldsymbol{(1)} = 2\mu \cdot e(2) \cdot \hat{\boldsymbol{y}}\boldsymbol{(2)} + 2\mu \cdot e(1) \cdot \hat{\boldsymbol{y}}\boldsymbol{(1)}$. Based on this rule, finally we have the converged coefficients

$$\hat{\boldsymbol{h}}(n) = 2\mu \cdot e(n) \cdot \hat{\boldsymbol{y}}\boldsymbol{(n)} + ... + 2\mu \cdot e(2) \cdot \hat{\boldsymbol{y}}\boldsymbol{(2)} + 2\mu e(1) \cdot \hat{\boldsymbol{y}}\boldsymbol{(1)}$$

where $\hat{\boldsymbol{y}}\boldsymbol{(1)}, \hat{\boldsymbol{y}}\boldsymbol{(2)}, ..., \hat{\boldsymbol{y}}\boldsymbol{(n)}$ are sine-like vector having exactly the same frequency but different initial phase and amplitude. Since the summation of sine functions, given variable initial phase and amplitude, is still a sine function, therefore the plot of converged filter coefficients $\hat{\boldsymbol{h}}(n)$ looks like a sine with same frequency as $y(n)$, which is f_0.

In the empirical section, question 4, we obtain zero padding filter coefficients when we increased the filter length from 10 to 100. Assuming that we get converged filter coefficients $\hat{\boldsymbol{h_{10}}}$ at length of 10, the error function is $e(n) = \hat{\boldsymbol{h_{10}}}^T(n-1)\boldsymbol{y}(n) = 0$ for any index n. By increasing the filter length, we add zero after the converged coefficients $\hat{\boldsymbol{h_{10}}}(n)$ and thereby create a new filter $\hat{\boldsymbol{h_{new}}}(n)$. Then the error function for the new filter is given by

$$e(n) = x(n) - \hat{\boldsymbol{h_{new}}}^T(n-1)\boldsymbol{y}(n) = x(n) - \hat{\boldsymbol{h_{10}}}^T(n-1)\boldsymbol{y_{10}}(n) + \hat{\boldsymbol{h_{zero-padding}}}^T(n-1)\boldsymbol{y_{left}}(n)$$

where $\boldsymbol{y_{10}}(n), \boldsymbol{y_{left}}(n)$ denotes the vector consisting of first 10 elements and other elements. The last term is 0 because of the zero-padding. Since the solution had previously converged to an optimal solution for $\hat{\boldsymbol{h_{10}}}$ we still have that $x(n) - \hat{\boldsymbol{h_{10}}}^T(n-1)\boldsymbol{y_{10}}(n) = 0$. Thus, the error function of new filter is zero at the beginning ($\Rightarrow \hat{\boldsymbol{h}}(n) = \hat{\boldsymbol{h}}(n-1)$) and the zero coefficients are kept unchanged after training. This is the reason we obtained the filter of a zero-padding shape when increased filter length.

C code:

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

This code is able to work with approximately 220 taps.