SSY130- Project 2
Adaptive Noise Cancellation

Group 8

Kashif Shabir
Haitham Babbili
Muhammad Haris Khan

December 17, 2019

Empirical Section

1.

We found the following results for the Broadband noise after using 300 taps for the filter

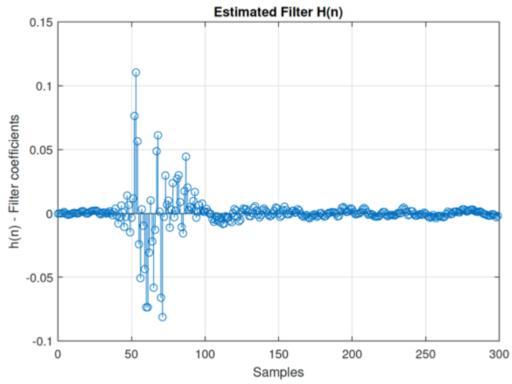


Figure.1 LMS estimated filter, Discrete Time Coefficients

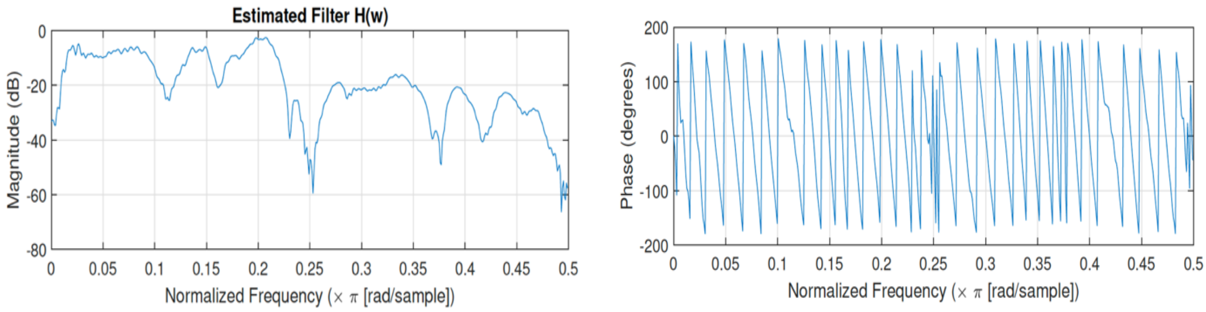


Figure.2 LMS estimated filter, frequency response

2.

(a) The error start escalated significantly by adding disturbances or by change the character of the input signal, we start hearing noise without updating the filter. It means, this effect can be seen in figure (3) below.

This is because the real part of the channel h(n) changed and our last estimated channel $\hat{h} = (n)$ does not correspond to this channel anymore since we are not updating our filter. In order to see that effect, we put a notebook between the speaker and DSP-kit at t ≈ 164s, the DSP-kit isn't able to filter the noise correctly anymore.

Figure.3 The effect of noise when input signal is changed but filter coefficient remains the same

(b) The noise will be filtered independently as long as there is no change in the channel or we have a good channel prediction $\hat{h}(n)$.
Moreover, from equation

$$e(n) = s(n) - \varepsilon(n)$$

we know that a perfect filtering will result

$$e(n) = s(n)$$

However, when we increase the volume of the speaker, we also increase the amplitude of the sound s(n) and therefore also the error e(n) amplitude as well. Actually, this does not mean we have worse filtering. If we compare with question (a), in both situations the error increases, but in here the filtering quality remains the same. Figure (4) below shows us the effect of decreasing the volume at t=40s and increasing again at t=50s.

Figure.4 The effect of increasing and decreasing the volume throw the time

(c) We allowing filter coefficient to converge to a good minimum error value and long there is no change in the channel corresponding to the noise source, changing the signal source will successfully keep removing the disturbance signal. This happens because we don't care about the sound *S* either it's transmitted or not. If we have a good estimate of the channel, then we can remove the noise successfully. Figure (5) elaborates what happens when we change the position and volume of the signal source. As discussed earlier, keeping the quality of filter remain same, error increased.

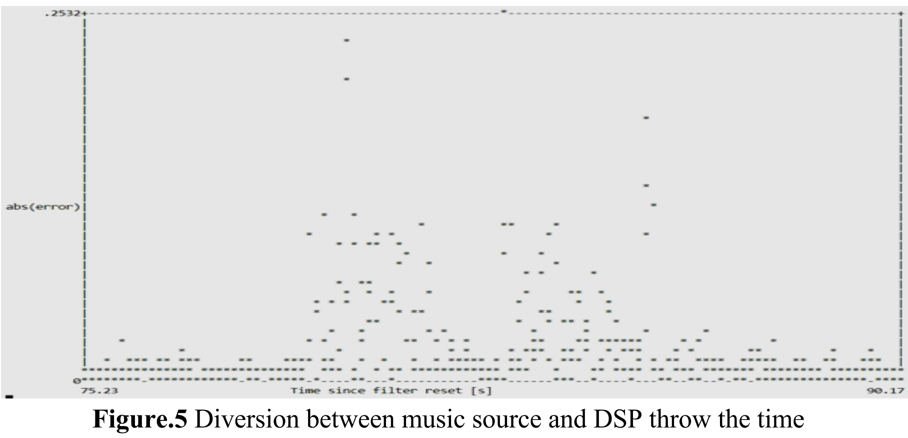
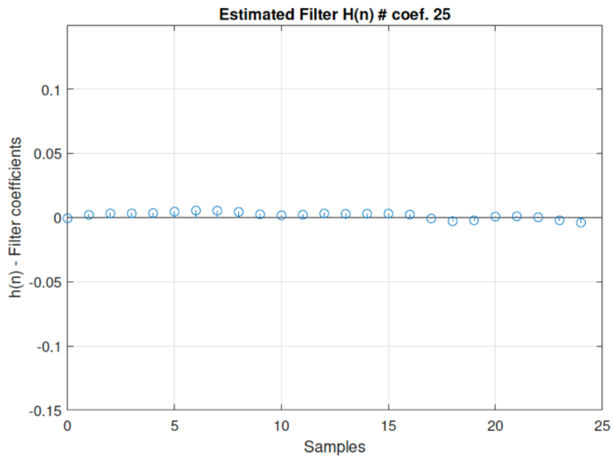
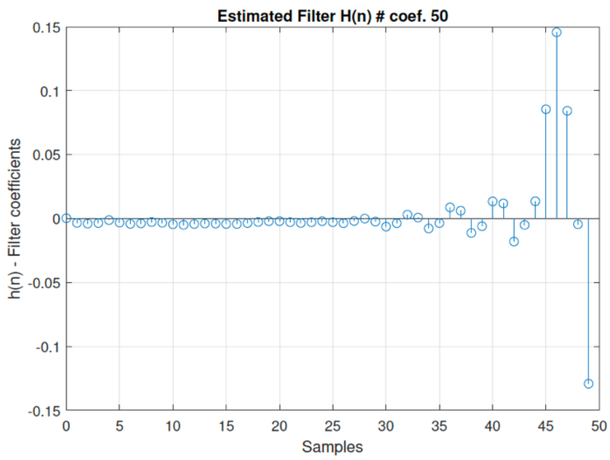


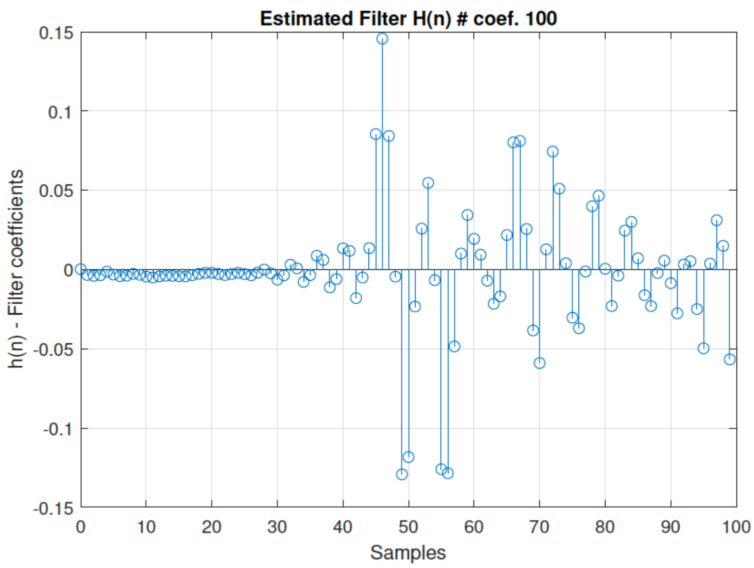
Figure.5 Diversion between music source and DSP throw the time

3.

During experiment we observed the filter doesn't behave equally for different lengths. Our conclusion is at 100 and even at 50 we get average performance, but the performance became bad lower than 45, so 45 is critical length for the filter (see figure below). Depending on the question 1 the maximum number of possible coefficients is 300. As the coefficient being updated by LMS algorithm all the time while the experiment is running, we should not reset the coefficients while reducing to the maximum length to 25. Hence, the coefficient always converge to global minimum because it's a convex optimization problem.
Under similar experiment conditions, we still got the same performance for similar filter coefficients, so that as much as we reset the coefficient to zero.
The start value to the coefficients of the filter at 45 are very close to zero because they are related to the accumulated time delay while transmitting and receiving the sound. Because of that, all the coefficients smaller than 45 will give a very bad filtering, hence all coefficients will be very close to zero due to the time delay and will not be able to represent correctly the real channel h.







4.

Our observations that the shape of the coefficients remain the same when we are decreasing number of elements from 100 to 10. That we can see it clearly in the figure below.

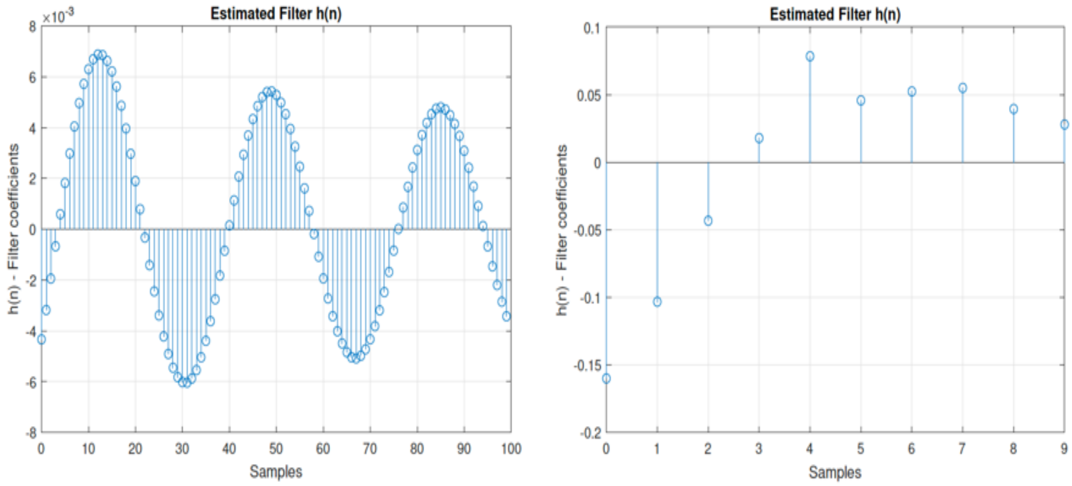


Figure (7) show the shape of decreeing value for coefficients

We repeat the test by increasing the coefficient's value from 10 to 100 obviously the amplitude and the phase remain the same and the additional coefficients added to the end with zero initial value. Based on that we don't have to reset the coefficients value when we change the length of the filter. The coefficients will converge under specific convergence conditions with some delay since it belongs to the convex optimization problem.

According to the figure we see h_{sin} filter works well compared figure (7) until n = 0 and the results are quite better for coefficients greater than 10. The results for both filters are similar having very low error, even the filter coefficients are different. This means that h_{sin} is not unique, since we are probably using more parameters than necessary.

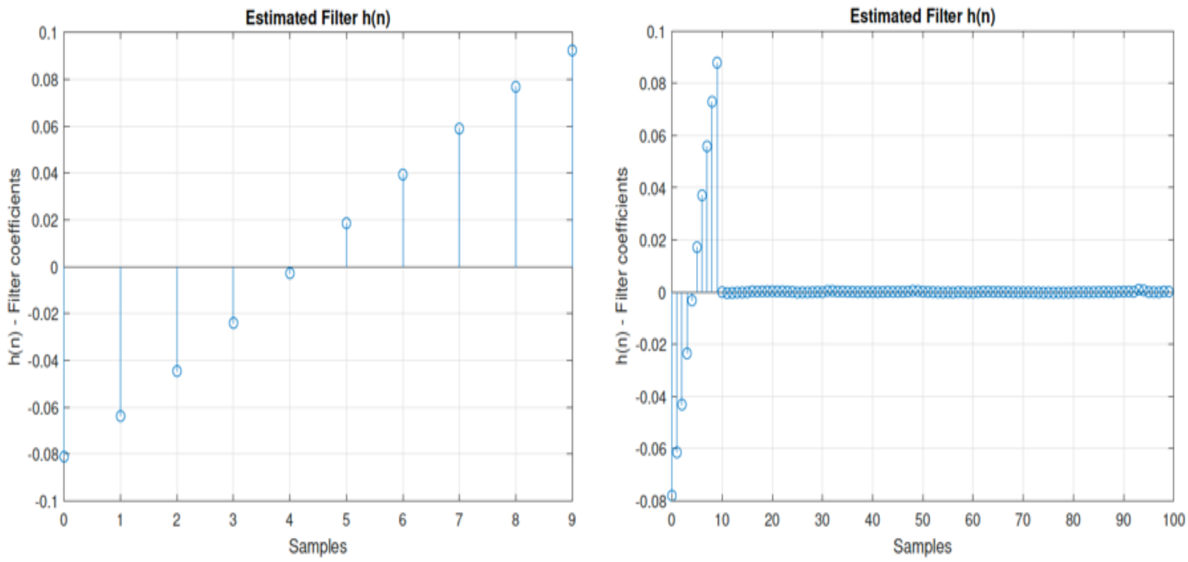
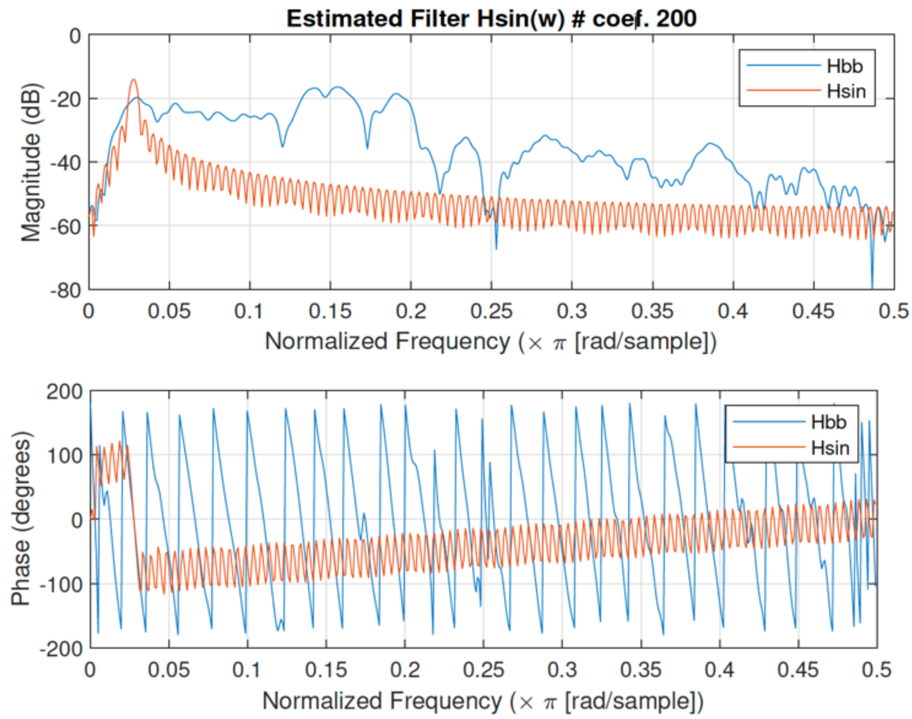


Figure (7) show the shape of increasing value for coefficients

5.

We saw that h_{sin} is not able to reach the broad-band disturbance, because the noise can be heard with the music when we put the left speaker to our ear. On the other hand, the h_{BB} filter does attenuate the sinusoidal noise disturbance. This is because h_{sin} is optimized to remove only the noise in frequency (440 Hz), while hBB was trained to remove noise of a large frequencies range that compose the broadband signal, which includes the 440 Hz. As can be seen in figure below, both hBB and hsin have similar magnitude and phase response for 440 Hz and that is why hBB works well when applied to the sinusoidal noise. On the other hand, it is clear that hsin has not adjusted its magnitude and phase response for other frequencies so it will not be able to filter well signals of other frequencies other than 440Hz.



6.

The signal from cellphone speaker which is large enough to saturate the channel or we can say in other words it will clip the channel. Non-linearity has been start observing in channel just because of this clipping, so convolution and linear operations can no longer be applied. The coefficients of the estimated saturated channel $h_{BB,sat}$ change completely from the not saturated channel hBB. The LMS algorithm is trying to minimize the error, but because of non-linearity, error results will always be very high.

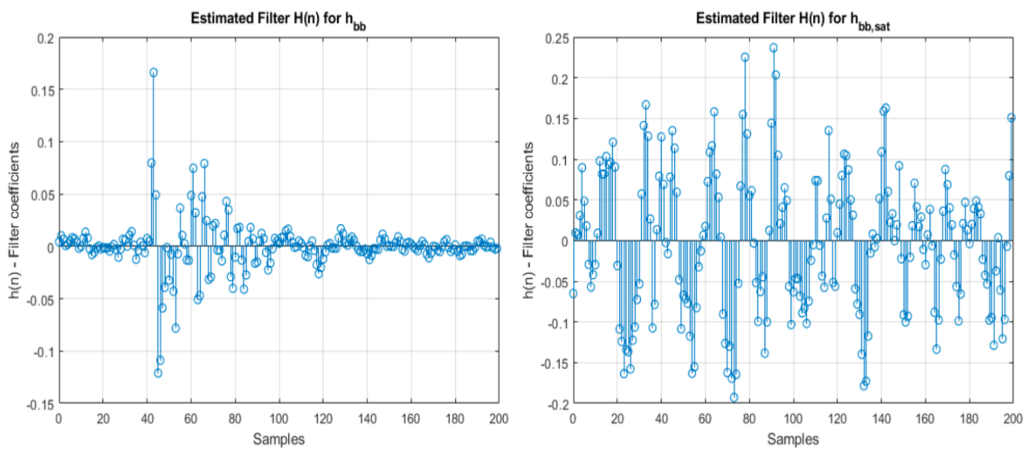


Figure h_{bb} not saturated Vs h_{bb} saturated

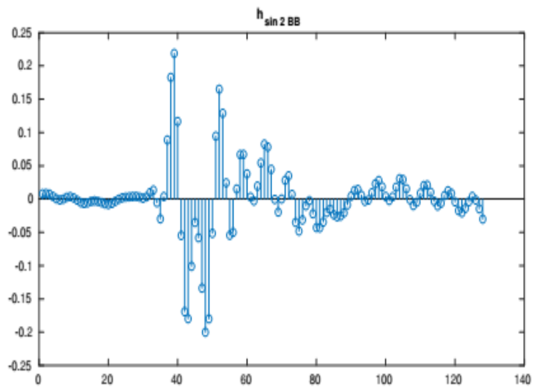
Analytical section

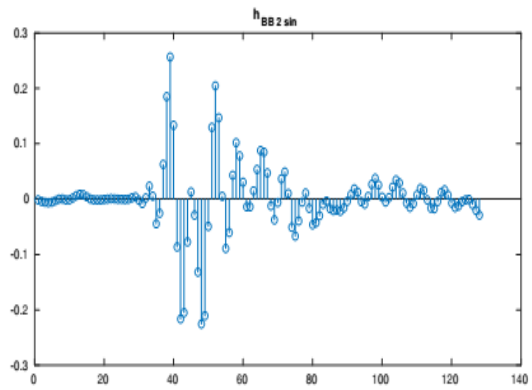
1.
a- The step size of the filter defines the convergence rate of the filter. If step size is large, the filter will converge fast because it takes less time to achieve local optima and vice versa. The Eigen values of the auto-correlation matrix of received signal determine the choice of mu
b- When we use large step size the estimated channel may diverge because the error function will start diverging from any specific point because we are taking less filter coefficients and each iteration actually depends on two points: the previous error signal and mu. If selected mu is too large or error increases if we take large step size then LMS algorithm will diverge.
c- After achieving convergence, the step size affects the filter performance as well. Since step size leads to the rate of the convergence, greater the step size faster the rate of convergence and vice versa. However, larger step size give us a bigger zone and vice versa. After achieving convergence, if we the filter behavior lies within those convergence zones then it will show similar performance else it will start diverging again.

2.
Case I: Decreasing filter length
In case of sinusoidal noise there is no difference in error performance when we decrease filter from 100 to 25 however, the error increases when we take broad band noise. Therefore, the optimal length of the equalizer filter must be between 100 to 150 in order to compensate channel effects.

Case II: Increasing filter length
When we use broad band noise there is no difference of changing the length of the filter if we reset the filter taps or not. However, filter showing same performance since LMS objective isn't unique, this is because this approach is already close to initial LMS performance.

3.





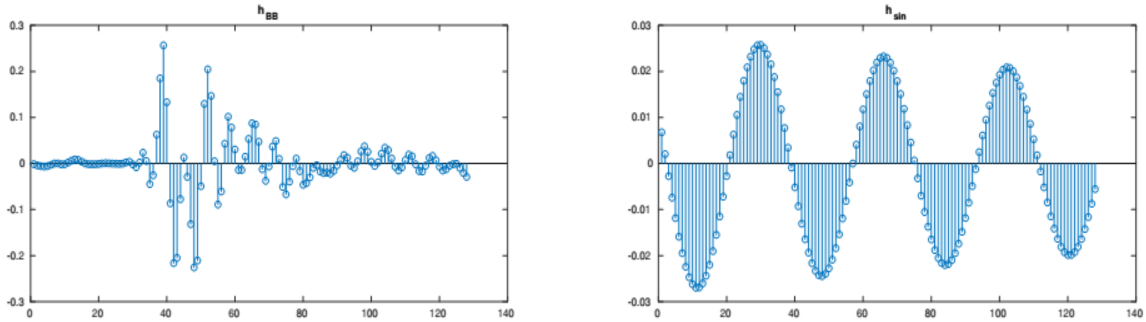


Figure 11: Stem plot of coefficients for the four cases, from top left to bottom right: h sin2BB, hBB2sin, hBB, hsin

Looking carefully the figure 9, we can conclude that filter coefficients from hbb and hsin2bb are similar and from hsin to hbb2sin are different.
Hbb and hsin2bb are similar because when we change from broadband to sinusoidal noise the filter was already tuned for frequency of 440Hz which was already present in broadband. That is why there is no change from H_{bb} and $h_{sin\to bb}$.

On the other side, there will be no different between $h_{sin\to BB}$and h_{BB} since filter coefficients update to match the magnitude and phase of other new frequencies given by the broadband noise.

If we wait more enough time $h_{BB\to sin}$will not approximate hsin. After all, $h_{BB\to sin}$looks similar to hBB and the error is already very small for the new sinusoidal noise, then the error gradient will also be very small, and the coefficients will be almost not updated.

4.

We observe from the figure (9) that $Hsin(f0) = 1{:}43\angle 1{:}9$ and $H(\mathrm{f}_0)_{\mathrm{BB}} = 1{:}38\angle 12.2$. As we know H_{BB} and H_{sin} are trained with noise that includes sinusoidal signal contain (f_0=440Hz), our mission is to try to match the magnitude and the phase of $\widehat{H}$ with the real channel H for this frequency. Based on that we will get similar phase and amplitude for $H(f_0)_{sin}$ and $H_{BB}(f_0)$.
Filter's phase and magnitude will be different for the frequency different than f_0. that because of, we trained H_{sin} to cancel a particular frequency while H_{BB} is trained to cancel a large frequency range contained in the broadband signal.

5.

We need to cancel a sinusoidal disturbance of the FIR filter to determine the number of filter coefficient that we need.
It's useful to solve this issue in frequency domain using the following minimization problem

$$\hat{H} = arg\ min\ \frac{1}{\omega} \int_0^{\omega s} |H(\omega)|^2 S_{SS}(\omega) + (|H(\omega)|^2 - |\bar{H}(\omega)|^2) S_{yy}(\omega)\ d\omega$$

Where $S_{SS}(\omega)$ and $S_{yy}(\omega)$ are the frequency responses for input and output signals.

For sinusoidal disturbance input $S_{yy}(\omega)$ will be a complex signal and only non-zero at $\omega = \pm 2\pi f0 + k\omega_s$. Which simply means the estimated channel is equal to the actual channel at $f0$ like

$$H(\omega_0) = \hat{H}(\omega_0)$$

In order to find FIR filter coefficients in order to map them on to the complex plane since our $\hat{H}(\omega_0)$ is a complex signal. The DFT can be written as

$$H(k) = \sum_{n=0}^{N-1} h(n)\ e^{-j\frac{2\pi kn}{N}}, \quad n = 0, 1, ..., N-1$$

The FIR coefficients can be calculated using the above equation. If we take only one coefficient that will a real number however, if we take more than 1 then H(k) will become a complex number which is the demand of the required filter in order to map the complex plane so that we can get $H(\omega_0)$.

6.

The plot in figure 4 in the empirical part shows that the FIR filter coefficients shows a sine behavior and this can be explained by using the LMS algorithm as follows:

$$e(N) = x(N) - \hat{\boldsymbol{h}}(N)\boldsymbol{y}(N)$$
$$\hat{\boldsymbol{h}}(N+1) = \hat{\boldsymbol{h}}(N) + 2\mu\boldsymbol{y}(N)e(N)$$

Looking that equation telling us that the next filter coefficient depends on the error function of the previous coefficient with some scaling factor with different phases. Similarly, for a particular frequency like f_0 we will get sum of sinusoids with that same frequency.

Appendix – LMS C Code

Listing 1: Noise cancelling LMS algorithm (Can handle 425 coefficients)

```
int n ;
      float32_t y_u[lms_taps] ;
      for ( n=0; n<block_size ; n++){
        arm_dot_prod_f32( lms_coeffs , lms_state+n, lms_taps , xhat+n ) ;
        e[n] = (x[n] - xhat[n]) ;
        arm_scale_f32 ( lms_state+n , e[n]*(2*lms_mu) , y_u , lms_taps ) ;
        arm_add_f32 ( y_u , lms_coeffs , lms_coeffs , lms_taps ) ;
     }
```