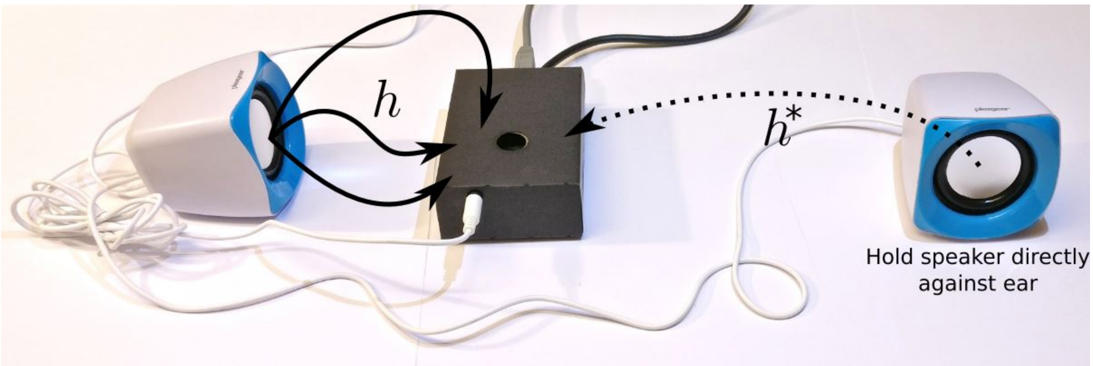
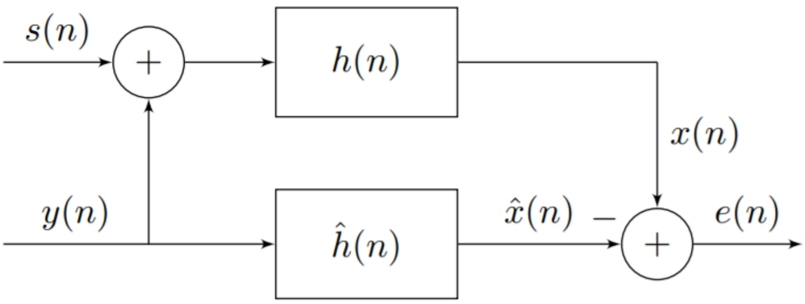
SSY130 - Project 2
Adaptive Noise Cancellation

Diego Jauregui   Lucas Rath   Mohammad Shaikh   Sondre Wiersdalen

December 18, 2018



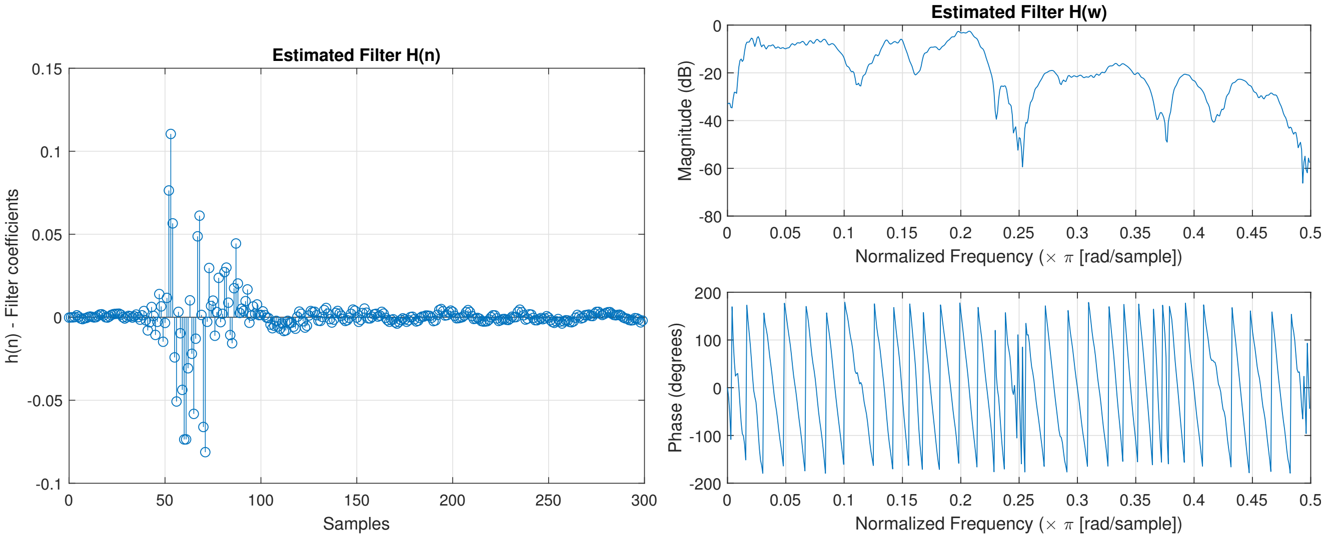


Empirical section

(1) The convergence rate is extremely sensitive to the chosen step-size. On one hand, if the step size is too small, the filter takes a long time do converge. On the other hand, if we select a high step-size then the error function starts to diverge from to local optima and the filter coefficients go to infinite.

---

(2) Using a number of 300 taps for the filter we obtained the following results for the Broadband noise case:



(a) Discrete time coefficients

(b) Frequency response

Figure 1: *LMS estimated FIR filter*

---

(3) If we suspend the DSP-kit and the speakers in free space we expect the physical channel to be almost constant over frequency. However, there are many channel effects than are being captured by the LMS filter. The first effect is the time delay corresponding to the sum of the time that takes for the sound to travel from the speakers to the DSP-kit box and all the latency and delays while emitting and measuring the signals. As can be seen in figure 1a, the filter coefficients of $\hat{h}(n)$ is almost zero until sample 50. This is represented by a linear phase shifting in frequency domain, as seen in figure 1b and is given by the following equation:

$$x(n-50) \rightarrow e^{-j\,50\,(w/f_s)} X(w) \tag{1}$$

The second effect that is being captured is the bandpass character of the acoustic system, which has a small gain for low and high frequencies. This is primarily due to the loudspeakers (limited to signals > 50 Hz $= 0.006\pi$ rad/sample) and microphone (limited to signals < 8 kHz $= 0.5\pi$ rad/sample). This low gain regions can be verified in figure 1b.

Third and last, even when making experiments in a very quiet and large room, we have effect of echos in the surroundings of the DSP-kit box. As an example, if we consider $\hat{h} = [1,\ 0,\ 0,\ 0.5]$, this is the time-domain representation of a channel with a direct component and an echo with half the amplitude after 4 samples. The DFT of $\hat{h}$ can be seen in 2 and it is pretty similar to what we obtained in 1b.

In conclusion, adding all this effects together, it is quite of expected to get the frequency response as seen in 1b.

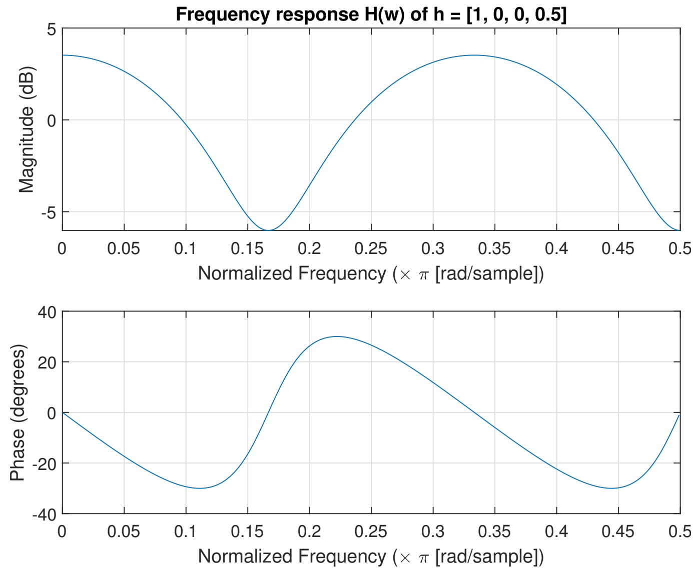


Figure 2: *Frequency response of the filter h = [1, 0, 0, 0.5]*

---

(4) **(a)** If we introduce disturbances or change the character of the input signal and do not update the filter, the error increases substantially and we start to hear the noise. This effect can be seen in figure 3. At $t \approx 165s$ we have introduced a book between the speaker and the DSP-kit. This happens because the real channel $h(n)$ changed and our last estimated channel $\hat{h}(n)$ does not correspond to this channel anymore, and therefore the DSP will not be able to filter the noise correctly anymore.

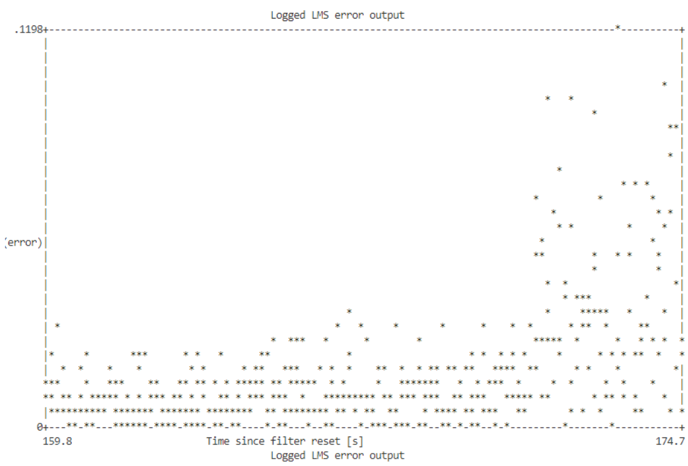


Figure 3: *Error when input signal is changed without changing filter coefficient*

(b) As far as we do not change the physical channel and we have a good channel prediction $\hat{h}(n)$, the noise will be filtered independent of the noise or sound amplitude. However, as can be verified in equation (3) in the assignment description, a perfect filtering will result $e(n) = s(n)$. As we increase the volume of the speaker, we amplify the sound $s(n)$ and therefore also the error $e(n)$. Clearly, this not implies that we have worse filtering. In comparison to question a), in both situations the error increases, but in this experiment the filtering quality remains the same. Figure 4 shows the effect of decreasing the volume at t=40s and then increasing again at t=50s.

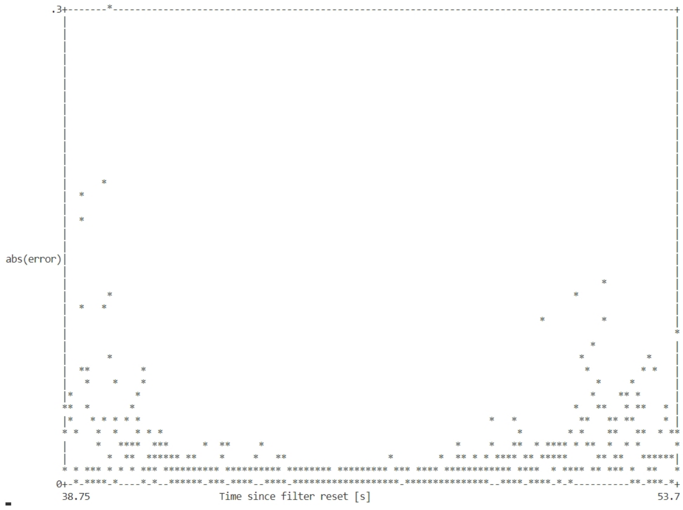


Figure 4: *Volume high at* $t \approx$ *38s, then volume decreased at t=45s and then increased again at t=50s*

(c) As far as we do not change the physical channel corresponding to the noise source, and we let the filter coefficients to converge to a good minimum error value, changing the signal source will successfully keep removing the disturbance signal. This happens because we do not care about the sound s is being transmitted. If we have a good estimate of the channel, in which the noise is being transmitted, then we can remove the noise successfully. Figure 5 shows what happens when we change the position and volume of the signal source. As discussed in the last item, the error increased, but the filtering quality remains the same.

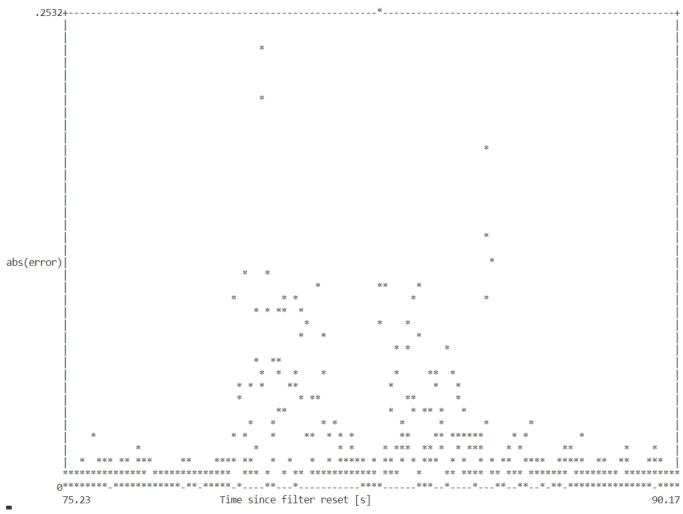


Figure 5: *Music source s(n) far away from DSP at* $t \approx$ *75s, then distance decreased at t=80s and then increased again at t=90s*

---

(5) The maximum number of possible coefficients is 300 (see question 2). While running this experiment we did not have to reset the coefficients when reducing from the maximum length thorough 25 because the coefficients are being updating by the LMS algorithm all the time. Since the optimization problem is ideally convex, the coefficients might always converge to the global minimum. Therefore, even if we reset the coefficients we obtain at the end very similar filter coefficients if, of course, if we keep the same experiment conditions.

During this experiment we have noticed that the filter does not work equally for all the filter lengths. We have observed a good performance with 300, 100, a decent performance with 50 and a very bad performance below 45 coefficients. We concluded that the critical length for bad performance is around 45. This length could be determined previously by looking at the plot of the estimated filter of length 300 in figure 1a. The initial 45 coefficients of the filter are very close to zero because, as explained in question 3, they are related to the aggregated time delay while emitting and receiving the sound. Therefore, any number of coefficients smaller than 45 will result in a very bad filtering, since 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.

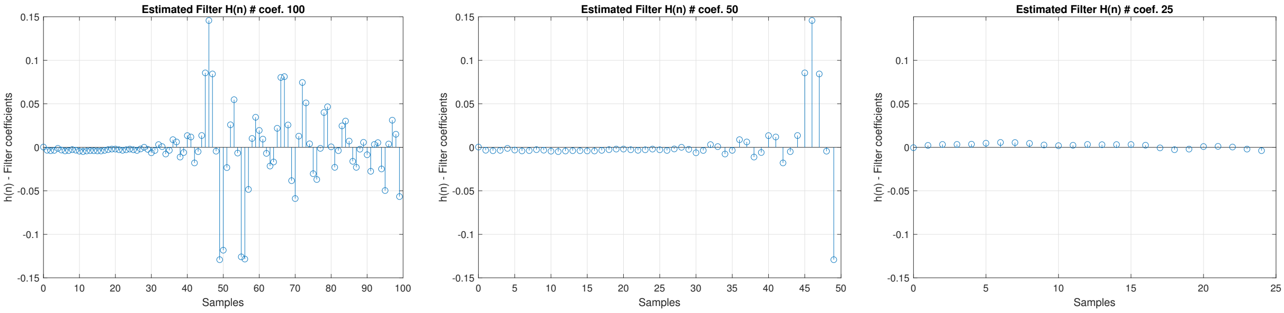


Figure 6: *Stem plot of filter of 100, 50 and 25 coefficients (from left to right)*

---

(6) Figure 7 shows what happens when we change from maximum length to 100 and then to 10 coefficients without resetting the filter. Indeed, the shape of the coefficients did not change when we decreased the number of elements. However, its magnitude was significantly scaled.

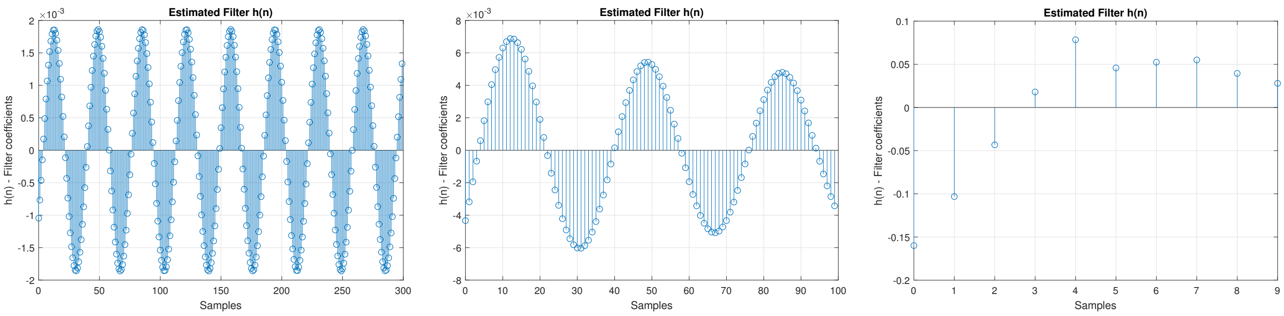


Figure 7: *Stem plot of filter of 300, 100 and 10 coefficients (from left to right)*

A similar test was made, but now starting with 10, then 100 and finally 300 coefficients, as seen in figure 8. Clearly, when increasing the number of coefficients, neither the magnitude or the shape of the stem plot changed and the additional coefficients added in the end stayed with its initial value 0.

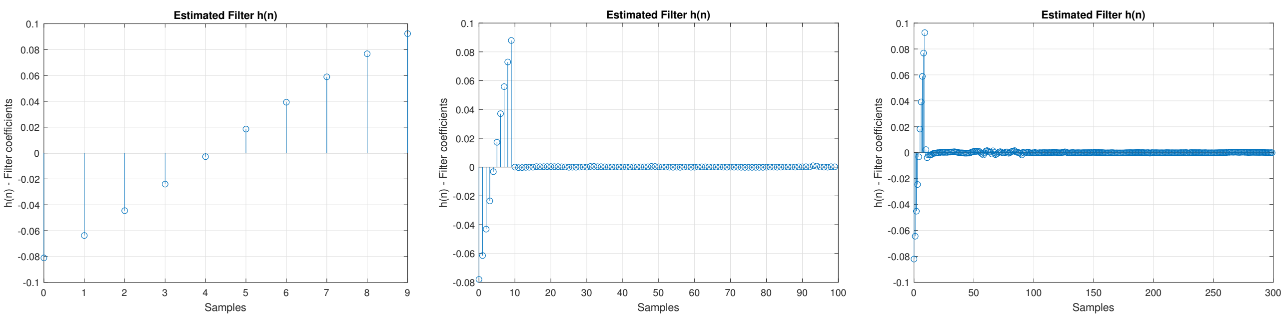


Figure 8: *Stem plot of filter of 10, 100 and 300 coefficients (from left to right)*

Accordingly, we do not need to reset the filter coefficients when changing the length. Since the optimization function is convex, the coefficients will most probably converge to the global optimum if given enough time and under certain convergence conditions.

The h_{sin} filter works well and results in a very low error for number of coefficients greater than 10. As can be verified when comparing figures 7 and 8 for n=100, both filters are different but result in a similar error, which is very low. This means that the solution (global minimum for the MSE) for h_{sin} is not unique, since we are probably using more parameters than necessary.

(7) The filter h_{sin} is not able to attenuate the broad-band disturbance, since the noise can still be heard along with the music when we put the left speaker to our ear. On the other hand, the h_{BB} filter does attenuate very well the sinusoidal noise disturbance.

This happens because h_{sin} is optimized to remove noise of only one frequency (440 Hz), while h_{BB} was trained to remove noise of a large range of frequencies that compose the broadband signal, which includes the 440 Hz. As can be seen in figure 9, both h_{BB} and h_{sin} have pretty similar magnitude and phase response for 440 Hz and that is why h_{BB} works well when applied to the sinusoidal noise. On the other hand, it is clear that h_{sin} has not adjusted its magnitude and phase response for other frequencies so it will not be able to filter well signals of other frequencies than 440Hz.

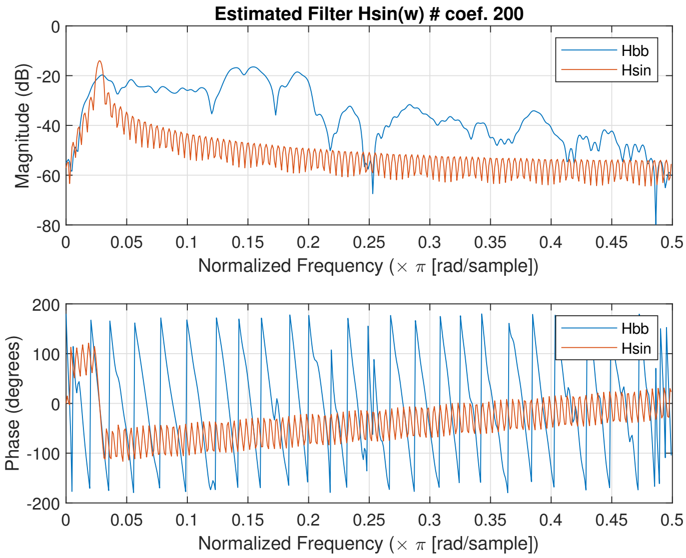
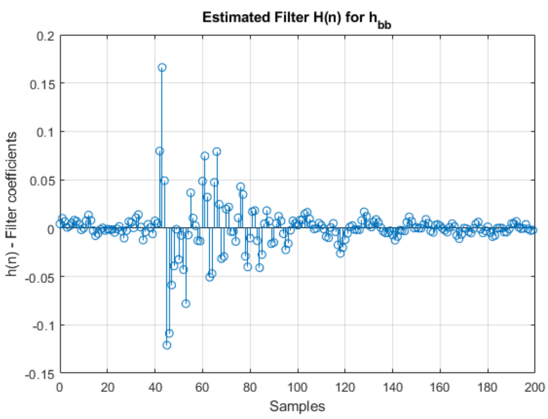


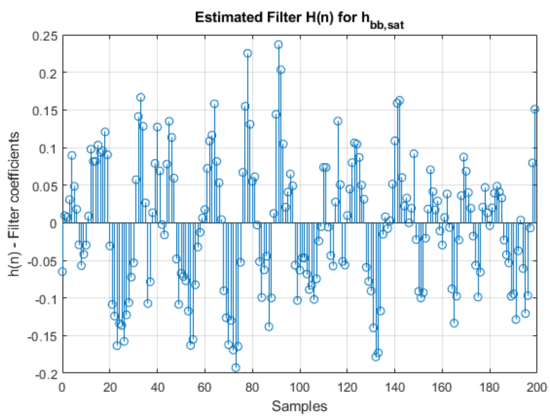
Figure 9: *Frequency response of H_{sin} and H_{BB}*

---

(8) The large signal coming out from the cellphone speaker will clip (or saturate) the channel. The clipping will introduce some non-linearities to the channel, so the convolution, Fourier transform and other linear operations can no longer be applied. The coefficients of the estimated saturated channel $h_{BB,sat}$ differ completely from the not saturated channel h_{BB}. The LMS algorithm is trying to minimize the error, but because of the non-linearities the operation will never converge and the result will be highly inaccurate.



(a) *h_{bb} not saturated*



(b) *h_{bb} saturated*

Figure 10: *Plot of the filter coefficients when speaker and cell phone used as a signal source respectively*

Analytical section

(1) A plot of the filter coefficients is provided below in Figure 11. By inspection the filter coefficients of h_{BB} and $h_{sin \to BB}$ are similar; and the coefficients of h_{sin} and $h_{BB \to sin}$ are different.

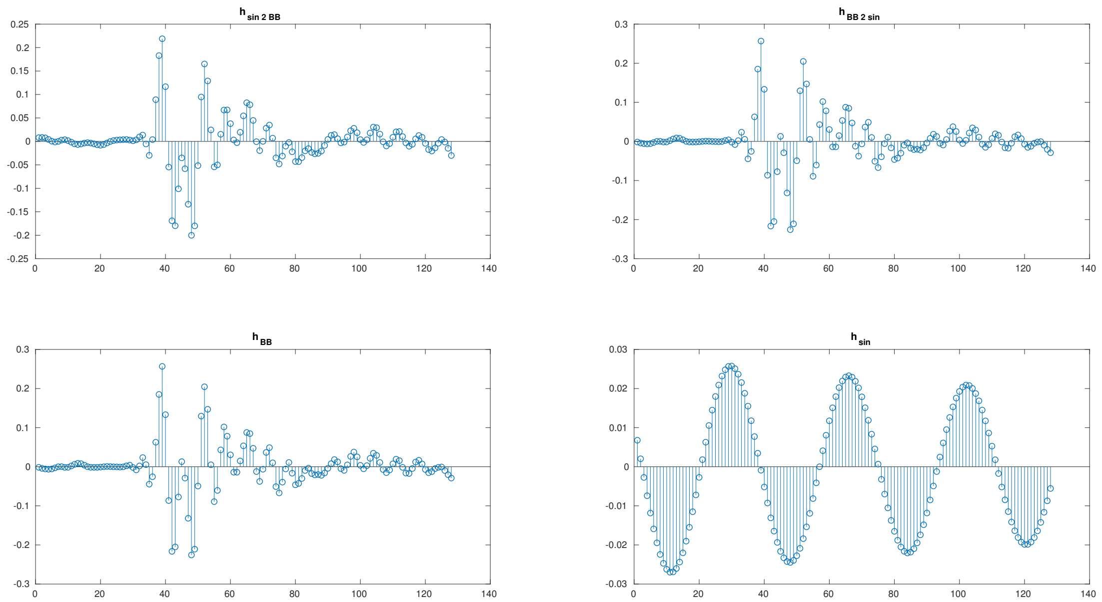


Figure 11: *Stem plot of coefficients for the four cases*

$h_{BB \to sin}$ is not changed when switching to a simple frequency disturbance because the broadband signal already includes sinusoidal signals with 440 Hz, so the filter coefficients were already trained to cancel the noise for this frequency. In frequency domain, we can say that H_{BB} has already adjusted its magnitude and phase to match the real channel for f=440 Hz, so the coefficients will not be updated in $h_{BB \to sin}$, since the cost function gradient will be already probably close to zero.

On the other hand, $h_{sin \to BB}$ is similar to h_{BB} because it will update its coefficients to match the magnitude and phase of other new frequencies introduced by the broadband noise.

If given enough time $h_{BB \to sin}$ will not approximate h_{sin}. Since $h_{BB \to sin}$ looks similar to h_{BB} 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.

---

(2) Since both H_{BB} and H_{sin} are trained with noise that includes sinusoidal components of f_0=440 Hz, the optimization procedure will then try match the magnitude and phase of $\hat{H}$ with the real channel H for this frequency. Therefore, we expect similar amplitudes and phases for $H(f_0)_{sin}$ and $H_{BB}(f_0)$. Both frequency responses can be seen in figure 9, which shows that: $H_{sin}(f_0) = 1.43\angle 1.9$ and $H(f_0)_{BB} = 1.38\angle 2.2$.

For frequencies different than f_0 the magnitude and phase of the filters should be different. They are different because H_{sin} is trained to cancel a specific frequency while H_{BB} is trained to cancel a large frequency range contained in the broadband signal.

---

(3) To determine how many filter coefficients an FIR filter needs to cancel a sinusoidal disturbance it's useful to look at the minimization problem in the frequency domain:

$$\hat{H} = arg \min_{\hat{H}} \frac{1}{w_s} \int_0^{ws} |H(w)|^2 \cdot \mathcal{S}_{ss}(w) + \left(|H(w)|^2 - |\bar{H}(w)|^2\right) \cdot \mathcal{S}_{yy}(w)\, dw \tag{2}$$

Where $\mathcal{S}_{ss}(w)$ and $\mathcal{S}_{yy}(w)$ are the spectrum of the signals $s(n)$ and $y(n)$ respectively. It is important to notice that $\mathcal{S}_{yy}(w)$ is a complex valued function and for the sinusoidal noise case, it will be nonzero only when $w = \pm 2\pi f_0 + \mathrm{k} w_s$.

This means that (2) is minimized when:

$$H(\omega_0) = \hat{H}(\omega_0) \tag{3}$$

Since $H(\omega_0)$ is a complex number the FIR filter must have enough filter coefficients such that the value of $\hat{H}(\omega_0)$ can be arbitrarily placed in the complex plane.

The definition of DFT is

$$H(k) = \sum_{n=0}^{N-1} h(n)\, e^{-j\frac{2\pi k n}{N}}, \quad n = 0, 1, ..., N-1 \tag{4}$$

The FIR filter with real valued coefficients is presented as follows

$$\hat{\boldsymbol{h}} = [\ \hat{h}_0 \quad \hat{h}_1 \quad ... \quad \hat{h}_N\ ] \tag{5}$$

The DFT of $\hat{h}$ for different filter lengths

$$\hat{H} = \begin{cases} \hat{h}_0, & N = 1 \\ \hat{h}_0 + \hat{h}_1\, e^{-j\pi k}, & N = 2 \end{cases} \tag{6}$$

By inspection of the results in (6) it can be concluded that: for $N = 1$, $\hat{H}$ is a real valued constant, hence it is not possible to arbitrarily place $\hat{H}(\omega_0)$ in the complex plane; and for $N = 2$ it is possible. It is therefore required at least two filter coefficients to cancel a sinusoidal disturbance.

---

(4) When using LMS to train the FIR filter for sinusoidal disturbances the plot of the coefficients (if using enough taps) will look like a sine. This can be explained by looking closer at the LMS algorithm in (7) and at the LMS update equation in (8):

$$e(N) = x(N) - \hat{\boldsymbol{h}}(N)\boldsymbol{y}(N) \tag{7}$$

$$\hat{\boldsymbol{h}}(N+1) = \hat{\boldsymbol{h}}(N) + 2\mu\boldsymbol{y}(N)e(N) \tag{8}$$

Since $\hat{\boldsymbol{y}}$ is a sine of frequency f_0, $\hat{\boldsymbol{h}}(N+1)$ will be composed by a sum of many sinusoids scaled by $2\mu e(N)$ with different phases. Clearly, a sum of sinusoids of frequency f_0 will result in a sinusoid of the same frequency f_0.

Appendix - LMS C Code

```c
    int k;
    //Temporary variable for the y vector update
    float32_t y_u[lms_taps];
    for(k=0; k<block_size; k++){
        // Dot product to calculate xhat = y*lms_coeffs
      arm_dot_prod_f32(lms_coeffs, lms_state+k,   lms_taps, xhat+k);
      // Error calculation, scalar sum
      e[k] = (x[k] - xhat[k]);
      // Scalar multiplication of a vector y_u = 2*mu*e[k]*y
      arm_scale_f32(lms_state+k, e[k]*(2*lms_mu), y_u, lms_taps);
      // Addition of 2 vectores lms_coeffs += y_u
      arm_add_f32(y_u, lms_coeffs, lms_coeffs, lms_taps);
    }
```

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