SSY130 Applied Signal Processing

Project 2: Adaptive Noise Cancellation

Group 9

YenPo Lin, Yu-hsi Li
Yun-Chung Chai, Wen Yu Yang

December 12, 2023

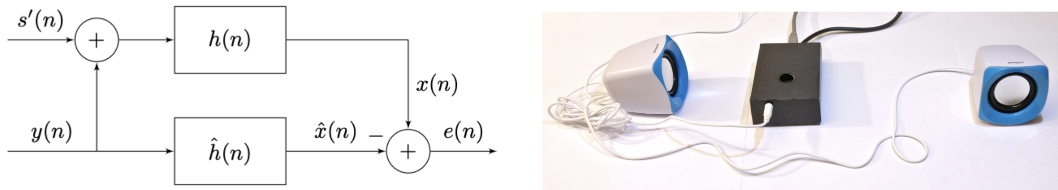

FIG. 1: real-world setup and signal diagram of noise cancellation setup

I. Empirical section

1

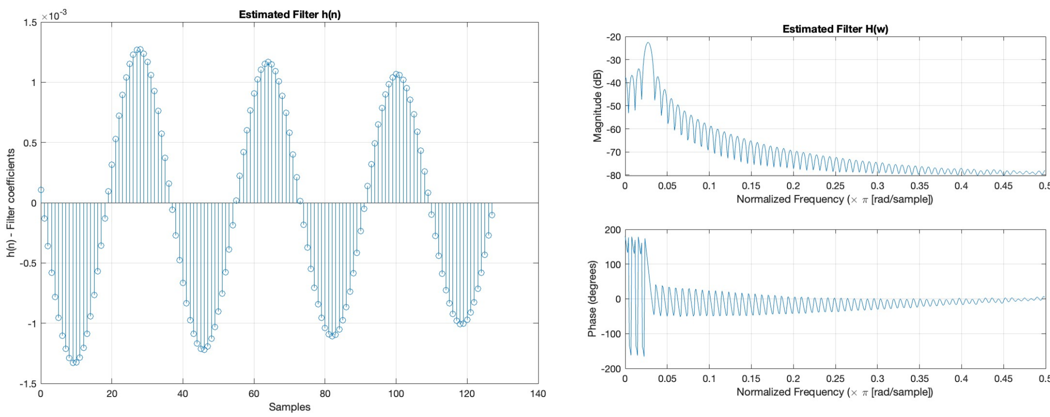

FIG. 2: sine noise case

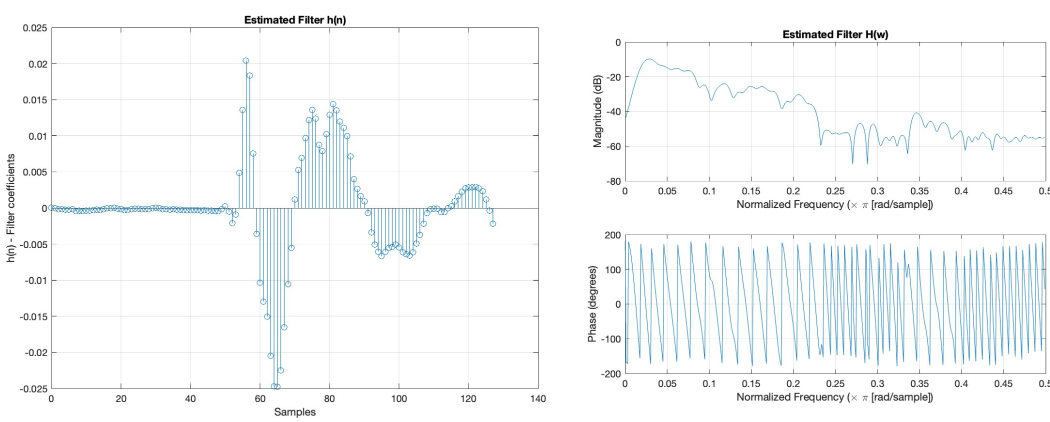

FIG. 3: Broadband noise case

2

(a) As we change the character of channel, the error output is observed to obtain a significant increment. The reason of it would be considered as when we training the filter using the initial character, the filter had converged and thus follow the character of the initial channel. Under this condition, if we modify the channel without training the filter again, instead, using the original trained filter will thus having this result that higher error outputs were being performed.

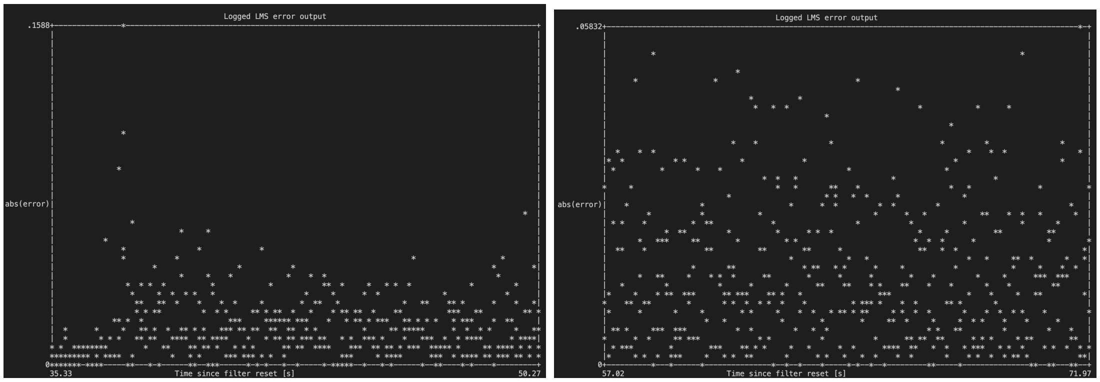

FIG. 4: with and without book in the middle

(b) The volume of the speaker outputs have a direct meaning to the amplitude of the disturbance. In this scenario, we tried to raised up the amplitude. The error performance have a trend of decreasing since the energy is raised leading to a lower error outputs while using a lower amplitude, in other words, a lower volume directly means have less energy when transmitting message implies that there is higher probability making the mistake. To conclude, the performance of the error have a positive relationship with the volumes.

Furthermore, the adjustment of the volume didn't modify the characteristics of the noise which in this case, the result is not similar to the one in Q2a.

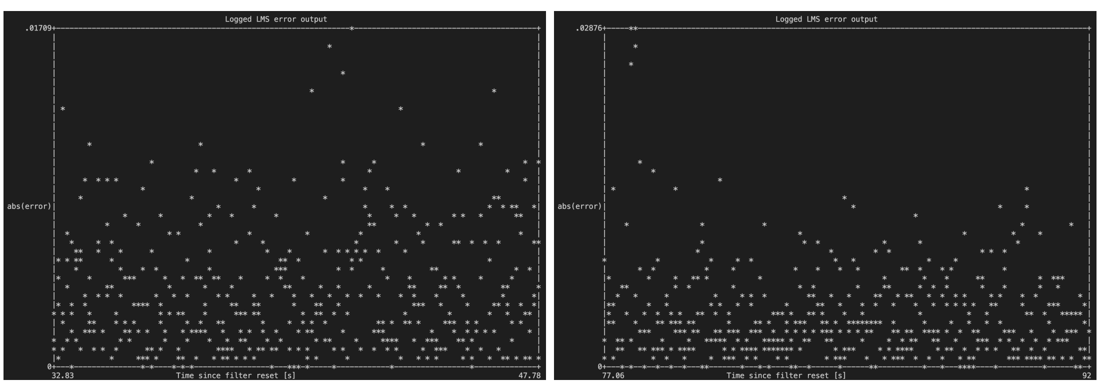

FIG. 5: lowering and raising the volume

(c) In our experiment, the filter had quite a decent result as we hearing the music without much noise. The output of DSP-kit is shown below. Further more, when we changed the position of the mobile, the filter seems to have the similar performance. This prove our assumption in subquestion 2a that if the character, more specifically, the environment noise is not severely modified, we would roughly have the same result with stable signal resources and moving signal source. However, there is one thing to mention, we do observed that the distribution of errors can varies according to the position of the mobile. To sum up, the moving mobile doesn't affect the performance of the filter but actually affect the patterns of error outputs.

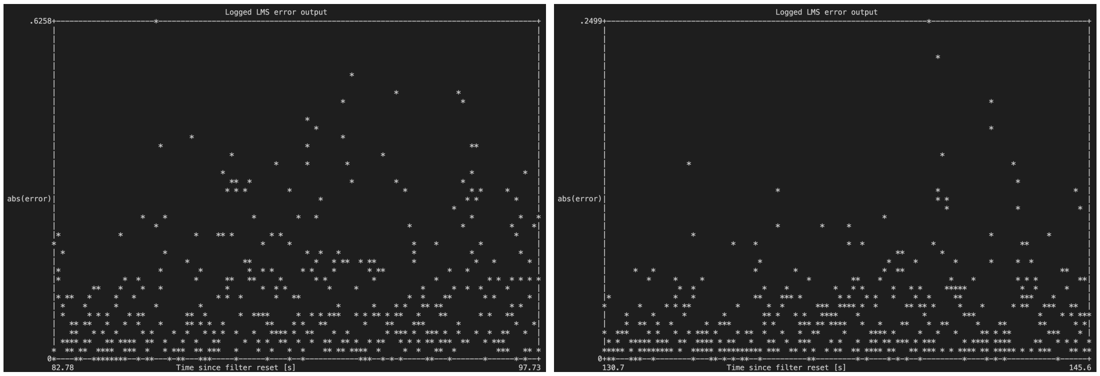

FIG. 6: fix position and move position

3

As the maximum length decrease, the performance will decrease obviously no matter we reset the filter coefficients. Because the adaption of the filter, the coefficients of the filter will converge to the same set of numbers.

However, as shown below, we do noticed that as we decrease the filter taps, the error output have a trend of increasing.

Besides, when modifying the tap to around 70, the error performance have a significant increment, this may imply 60 taps is a critical point when analyzing the result of our filter, to further prove on this, the stem plot in Q1 actually pointing out somewhere near 60 is a critical point to take a looks on .

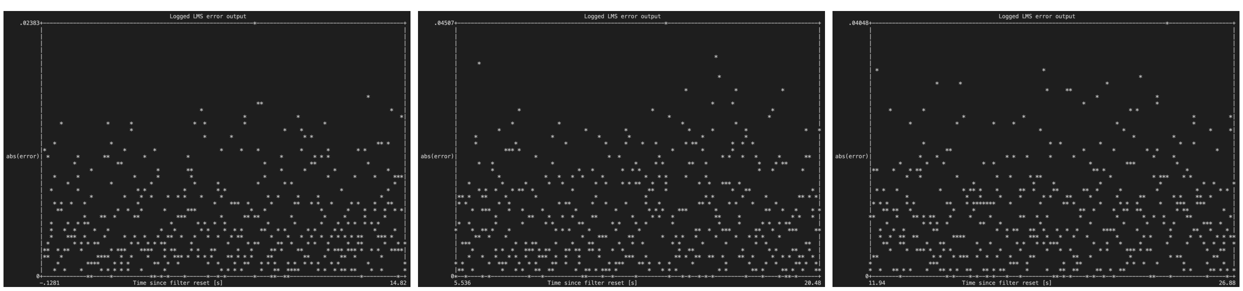

FIG. 7: Tap range from 100 - 50 - 25

4

As the experience result, the shape of the coefficients will not change when the number of elements decrease from 100 to 10. However, the amplitude of the coefficients have changed. When we go inversely from 10 to 100, the amplitude of the coefficients also change. However, the difference is that the remain number of coefficients set to 0 as the number of elements increase.

Besides, when altering the filter's length, it's not necessary to reset its coefficients. This is because the optimization function's convex nature implies that, with sufficient time and adherence to specific convergence criteria, the coefficients are likely to reach the global optimum.

We also discover that if we train the filter under the length of 100 and apply on length of 10, the performance would be arisen and meanwhile if we train the filter under the length of 10 and apply on length of 100, the result would be lower a bit but still decent.

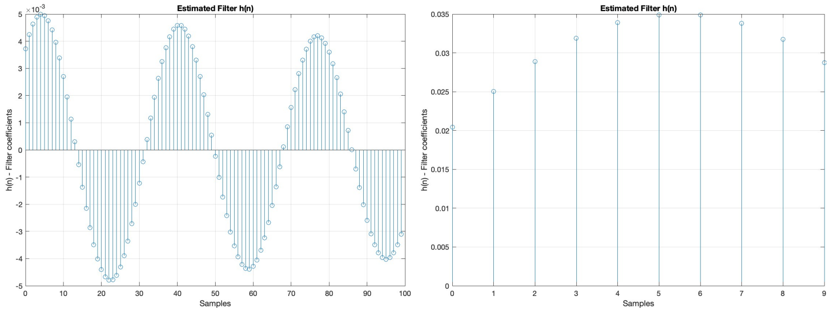

FIG. 8: decrease length from 100 to 10

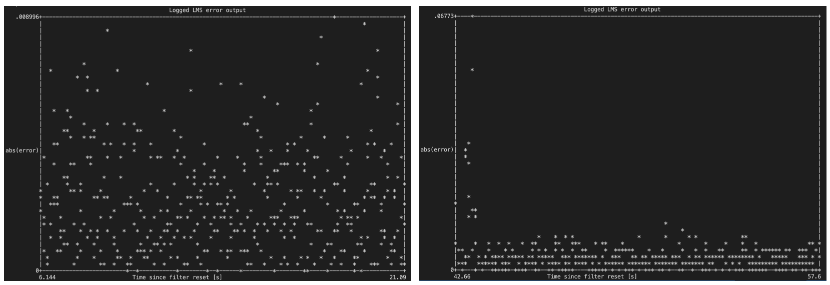

FIG. 9: error plot decrease length from 100 to 10

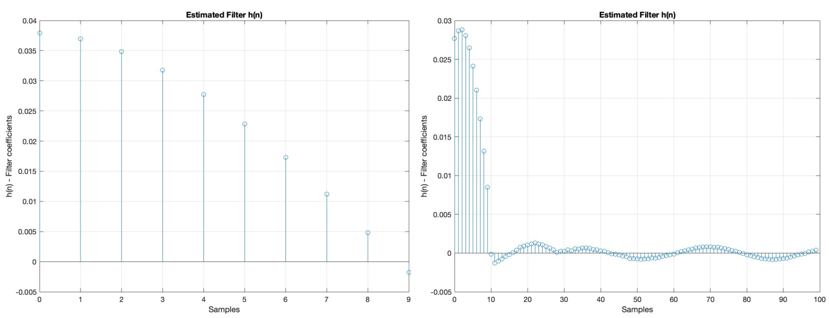

FIG. 10: increase length from 10 to 100

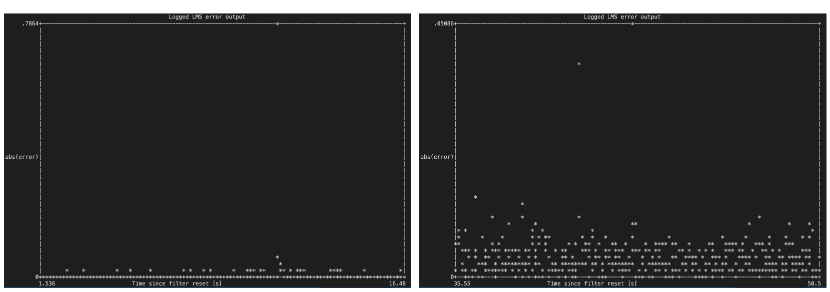

FIG. 11: error plot increase length from 10 to 100

5

We have seen that when training the hBB filter and switch to h_{sin}, the filter can actually deal with this modification. The reason why it happen is that the filter would try to eliminate multiple frequency of noise in the hBB filter and if switching to h_{sin}, the specific frequency is recognized by the hBB filter. Due to this reason, the performance is still remain decent and desirable. On the other hand, if we switch the scenario using the trained h_{sin} filter to hBB noises, the performance wouldn't be expected acceptable() since there are several frequency of noise in hBB noise is never being trained during the h_{sin} filter which lead to this situation. hBB has a characteristic that contains several frequencies of noises while h_{sin} noise has only one frequency of noise leading to these alternative result when switching these conditions.

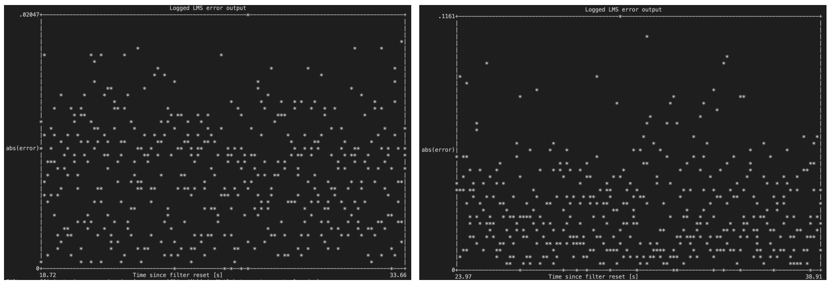

FIG. 12: h_{bb} in sinusoidal noise and h_{sin} in broadband noise

6

When high volumes are played on a mobile device, it can lead to channel saturation and subsequent clipping, introducing non-linearities into the channel, as depicted in This results in a notable difference between the coefficients of the saturated channel hBB,sat and the normal channel hBB.

The LMS algorithm, fundamentally designed for linear channels, struggles to adapt in the presence of these non-linearities. It attempts to apply linear filtering to the saturated, non-linear channel, but due to the channel's inherent non-linear nature, the algorithm fails to accurately estimate and correct for these non-linear characteristics, leading to ineffective performance and high error rates.

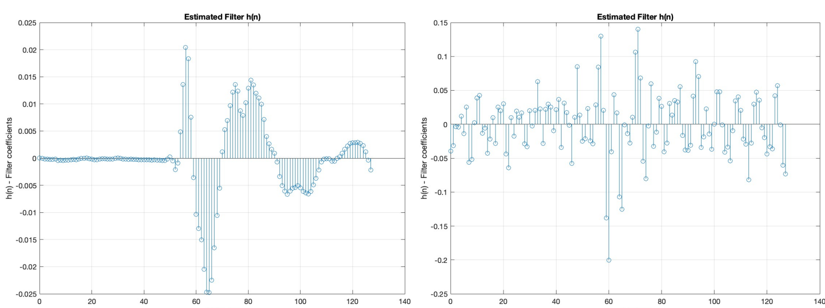

FIG. 13: original h_{bb} and $h_{bb,sat}$

II. Analytical Section

1

(a) The convergence rate is highly related to the choice of step size. If the size is too small, it needs to take more time to converge to optimal value and vice versa. (The Eigen-values of the auto-correlation matrix of the received signal determine the optimal choice of μ.) If a large enough step size is chosen, it is possible to lead the error function to overshoot the optimal values and start to diverge.

(b) Yes, the choice of step size affects the stability of the filter during the adaptation process, according to LMS, the updating of the filter only depends on the previous filter and updating part which includes μ, in normal adaptation, the filter will oscillate and converge toward optimal value. If the step size is too large, the oscillation will increase and affect the stability.

(c) Once convergence is achieved, the step size continues to impact the filter's performance to oscillate around the optimal value. A larger step size results in faster convergence but also provides a wider range in oscillation, while a smaller step size narrows it down. If the filter's behavior remains within the convergence zones after achieving convergence, it will exhibit similar performance. Otherwise, it will start diverging again.

2

Case 1: Decreasing filter length

When dealing with sinusoidal noise, reducing the length of filter from 100 to 25 does not affect the error performance. However, in the case of broad band noise, the error increases. The reason for this contrasting behavior lies in the requirement for compensating these two types of noise. For broad band noise, it needs lots of coefficients to capture the characteristic., while for sinusoidal noise only need small amount of coefficients are enough the estimate it. Base on previous section, the optimal length of the filter for compensating channel effects caused by broad band disturbances may fall within the range of 100 to 150.

Case 2: Increasing filter length

When dealing with broad band noise and increasing the length of the filter, whether the filter coefficients are reset or not does not result in any significant difference.

The reason behind this consistent performance lies in the nature of LMS algorithm. The LMS objective is not unique, meaning there can be multiple solutions that achieve similar performance. In this case, the filter's performance is already close to its initial LMS performance, and any changes in filter length or resetting of taps do not significantly alter its effectiveness.

3

If we train the filter with sinusoidal first then switch to broad-band disturbance, the coefficients will change because the original filter did not contain the other frequency component. Therefore, reversely, training with the broad-band noise first can cover the noise of the particular sinusoidal.

Switching to a simple frequency(sin) disturbance from broadband does not impact because the broadband signal already contains sinusoidal signals at 440 Hz. As a result, the filter coefficients were already trained to eliminate noise at this frequency. Conversely, training from sinusoidal and feed to broadband will update its coefficients to match the magnitude and phase of additional frequencies introduced by the broadband noise, and the graph will be eventually similar to h_{BB}.

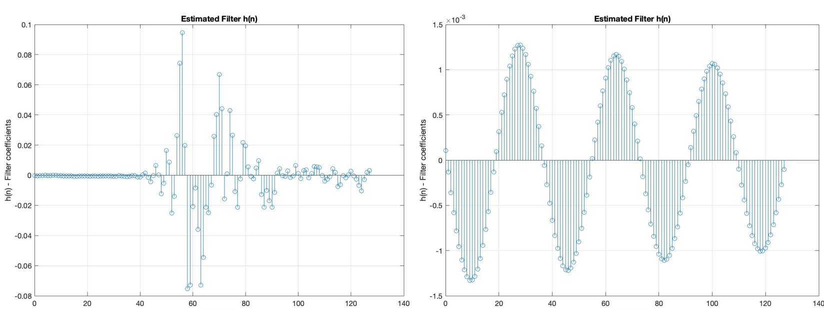

FIG. 14: h_{bb2sin} and h_{sin}

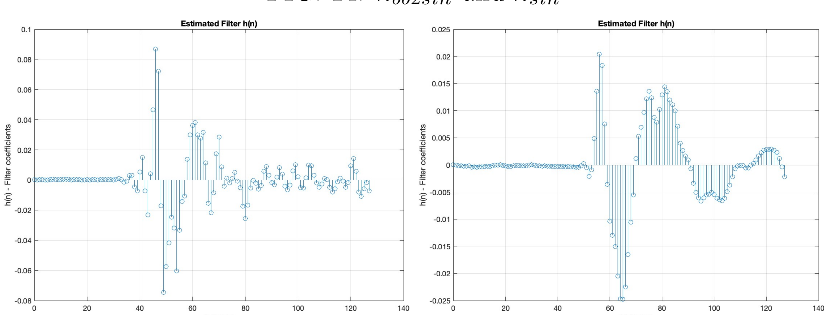

FIG. 15: h_{sin2bb} and h_{BB}

4

The results of H_{sin} and H_{BB} have only slightly different. The reason might be that H_{sin} system only needs to focus on the noise cancellation of a particular frequency (i.e. $f_0 = 440$ Hz), but H_{BB} was trained to cover a wider range of the noise frequencies. Therefore, H_{sin} should be more accurate than H_{BB}. $Hsin(f_0) = 0.047\angle 0.1$ and $H(f_0)BB = 0.32\angle 2.8$.

5

We need to cancel a sinusoidal disturbance of the FIR filter to determine the number of filter coefficient that we need. We can do it in frequency domain (from the lecture note)

To find the optimal filter, we can do it by minimize the term

$H_* = argmin\frac{1}{\omega_s}\int_0^{\omega_s}|H(\omega)|^2 \cdot S_{ss}(\omega) + (|H(\omega)|^2 - |H_0(\omega)|^2) \cdot S_{yy}(\omega)d\omega$

Where $S_{ss}(\omega)$ and $S_{yy}(\omega)$ are the spectrum of the signals $s(n)$ and $y(n)$ respectively

Assume that the channel is complex with amplitude and phase:$H_0(\omega_0) = A_0 e^{j\phi} = \sum_{k=0}^{M-1} h(k)e^{-j\omega_0 \Delta t}$

$$\begin{cases} Re. = h(0) + h(1)cos(\omega_0 \Delta t) + ... \\ Im. = -h(1)sin(\omega_0 \Delta t) + ... \end{cases}$$

Since the channel is complex, that mean one coefficient will be not enough to estimate the length of it (it will only have the real part), so we need 2 coefficient to do so:

$H_0(\omega_0) = H(\omega_0) = h(0) + h(1)e^{-j\omega_0 \Delta t}$

So theoretically, the minimum filter length $M \geq 2$, at that we will be able to cancel the sinusoidal disturbance with at 2 coefficients.

6

According to the LMS formula:

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

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

next filter coefficient depends on the error function of the current coefficient value. $\hat{y}$ is a sine of frequency f_0, so $\hat{h}(N+1)$ will be composed by many sinusoids function scaled by $2e(N)$ with different phases. Therefore, a sum of sinusoids of frequency f_0 will result in a sinusoid of the same frequency.

III. Appendix

```
for(int k=0; k < block_size; k++){
    float * y = &lms_state[k];

    arm_dot_prod_f32(lms_coeffs, y, lms_taps, xhat+k );
    e[k] = x[k] - xhat[k];
    for (int i=0; i<lms_taps; i++) {
        lms_coeffs[i] += 2 * lms_mu * y[i] * e[k];
    }

}
```

FIG. 16: LMS C-code