Design Review - EE2800

EE23BTECH11024 - Karthik Yadav
EE23BTECH11025 - Anantha Krishnan
EE23BTECH11026 - Hiba
EE23BTECH11027 - K Rahul

Choice of Algorithm

- The algorithm of choice is the LMS algorithm[1], more specifically, its normalized version, which is a very popular choice for ANC applications.
- Unlike in the case of the Weiner filter, the statistics of the signal(e.g, the autocorrelation function) is not known, hence we use the LMS algorithm. The LMS algorithm is explained in the next slide
- Although all these adaptive filters do assume stationarity, they perform well even for non-stationary signals as demonstrated experimentally.[2]

---

[1] Ying He,Hong He, Li Li, Yi Wu, Hongyan Pan, " *The Applications and Simulation of Adaptive Filter in Noise Canceling* ,".DOI 10.1109/CSSE.2008.370

[2] B. Widrow, J.M. McCool, et.al, " *Stationary and nonstationary learning characteristics of the LMS adaptive filter* ,". DOI 10.1109/PROC.1976.10286

LMS algorithm

- Let $\mathbf{x(n)}$ (in our case, it is the external noise vector, i.e, $\mathbf{w(n)}$) be the input signal, and the target signal be $d(n)$ (the target signal ideally is $\mathbf{v(n)}$, but we can use $\mathbf{s(n)} + \mathbf{v(n)}$ as well, because $\mathbf{v(n)}$, and hence $\hat{\mathbf{v}}\mathbf{(n)}$, are uncorrelated). Let the LMS filter weight vector be $\tilde{\mathbf{w}}\mathbf{(n)}$, which changes as a function of time (because we are performing adaptive filtering). Then, the update function[3] for the LMS algorithm is

$$\tilde{\mathbf{w}}(\mathbf{k+1}) = \tilde{\mathbf{w}}(\mathbf{k}) + \mu\mathbf{x}(\mathbf{k})(d(k) - \tilde{\mathbf{w}}(\mathbf{k})^H\mathbf{x}(\mathbf{k}))$$

- The convergence occurs for a particular range of μ in the MSE and mean sense. Too large a μ within the allowed range can make it oscillate, while too small a μ would take greater number of iterations to converge.

[3] Paulo S.R. Diniz, " *Adaptive Filtering Algorithms and Practical Implementation* ,"Springer, 4th Edition, pp. 137-141

NLMS algorithm

- For normal LMS algorithm, $0 < \mu < \frac{2}{tr(R_{xx})}$ guarantees convergence in the MSE sense, and hence, we would not be able to find the right range due to lack of knowledge of R_{xx}. Thus, we normalize the update function, so that

$$\tilde{\mathbf{w}}(\mathbf{k+1}) = \tilde{\mathbf{w}}(\mathbf{k}) + \left(\frac{\mu}{\|x\|_2^2}\right)\mathbf{x}(\mathbf{k})(d(k) - \tilde{\mathbf{w}}(\mathbf{k})^H \mathbf{x}(\mathbf{k}))$$

and thus, $0 < \mu < 2$

Practical Considerations

- As per the notation used in the Project Details document, the signal $\mathbf{s}(\mathbf{n})$ is considered to be uncorrelated with $\mathbf{w}(\mathbf{n})$, and hence, also $\mathbf{v}(\mathbf{n})$ and $\hat{\mathbf{v}}(\mathbf{n})$ as well. Although this may not perfectly be true, it is nevertheless a good engineering "approximation" taken to simplify, as the results are good enough (the SNR calculated is around 34 for the sample path given).

Things we were unable to figure out

- The ringing noise, at around 4 seconds into the audio recording, does appear, although it's amplitude is quite low as compared to the voice. We were unable to figure out as to how to remove that, or why it still persists.

Things we were unable to figure out

- The MSE vs iterations (convergence). plot is as follows:

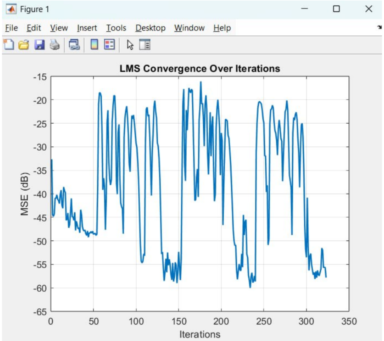

Figure: Convergence of LMS

- The LMS convergence seems to be pretty erratic - although it does finally settle to a value, it does not happen so without all the wild-looking oscillations before it. We are unable to figure out where this has come from.

Drawbacks and limitations of our code/approach

- If the total number of samples is not a multiple of the batch size, the remainder samples would not be processed, although that number would be extremely small ONLY if the batch size $\ll$ was the number of total samples, which is reasonable, given that the sampling rate is also quite high.

Future directions

- Currently, the filter indiscriminately removes all noise. We would have to implement partial suppression later. At the time of writing, partial suppression has not yet been explored, but we are looking at ways to achieve a better SNR for full suppression
- A promising candidate for good SNR seems to be the RLS implementation[4], which demonstrates superior performance as compared to the LMS implementation, although it is more hardware-intensive. But with the available computing power in today's hardware, this should not be a big concern.

[4] Ying He,Hong He, Li Li, Yi Wu, Hongyan Pan, " *The Applications and Simulation of Adaptive Filter in Noise Canceling* ,".DOI 10.1109/CSSE.2008.370