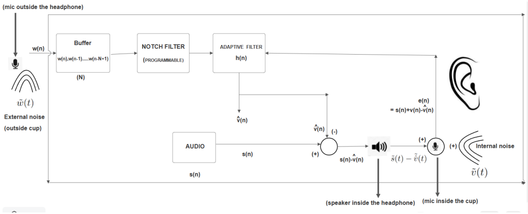
EE2800 Digital Signal Processing Course Project

Adaptive Filtering

KAUSTUBH KHACHANE - EE23BTECH11032
KILLANA JASWANTH - EE23BTECH11033
PRABHAT KUKUNURI - EE23BTECH11034
KUNAL THORAWADE - EE23BTECH11035

April 25, 2025

Block diagram , Problem Statement and Solution Approach



Problem statement:

- The headphone cup acts like a filter and the external noise signal w(n) gets filtered and enters the headphone cup as v(n) which acts as a noise while a person is using them. Now we have to implement a noise cancellation system which tries to model the filter characteristics of the headphone cup using an adaptive filter.

Key idea

- We will pre-subtract the estimated the noise $\hat{v}(n)$ from the sound we want to listen to s(n) so that it gets perturbed with the filtered noise v(n) and results in cancellation. Ideally we would like $v(n) - \hat{v}(n) = 0$. So that finally we hear s(n).
- Since v(n) is some filtered version of w(n) we can they both are correlated.
- The adaptive filter takes feedback to update its taps so that it can make the best prediction so we pass the error signal e(n) back to the filter.($e(n) = s(n) + v(n) - \hat{v}(n)$)
- To implement partial suppression mode we implement a notch filter which has programmable tonal frequencies, this notch filter removes the tonal frequencies from the external noise w(n),thus making it uncorrelated at those certain frequencies and the filter preserves these tonal frequencies after filtering as well.

Hardware specification of the headphone

- The w(n) is picked up the external microphone of the headphone, the error signal is picked up by the internal microphone of the headphones.

Design choices and justifications

Filter algorithm:

- **RLS:** Objective function : $\zeta(k) = \sum_{i=0}^{k} \lambda^{k-i} e(i)^2$

  $e(i) = s(n) + v(n) - \hat{v}(n) = s(n) + v(n) - h^T x(k)$

  minimising by differentiation wrt $h(k)$, the optimal weights for k^{th} iteration is $R_D^{-1}(k)p_D(k)$ where :

  $R_D(k) = \sum_{i=0}^{k} \lambda^{k-i} x(i)x^T(i) = \lambda R_D(k-1) + x(k)x^T(k)$ and $p_D(k) = \sum_{i=0}^{k} \lambda^{k-i} x(i)d(i) = \lambda p_D(k-1) + (s(k) + v(k))x(k)$

  Define $S_D(k) = R_D^{-1}(k)$ thus $S_D(k-1) = \frac{1}{\lambda}(S_D k - 1 - \frac{S_D(k-1)x(k)x^T(k)S_D(k-1)}{\lambda + x^T(k)S_D(k-1)x(k)})$

  In the k^{th} iteration filter coefficients are updated as $h(k) = S_D(k)p_D(k)$ and the estimated signal is given by $e(k) = s(n) + v(n) - h^T(k)x(k)$

Design Considerations:

- **Filter Order (N-1)**:The number of weights in the adaptive filter is N (RLS).
- **Forgetting factor(λ)**: $(0 << \lambda \leq 1)$.The smaller λ is, the smaller is the contribution of previous samples to the covariance matrix. This makes the filter more sensitive to recent samples, which means more fluctuations in the filter co-efficients.
- **Initialization scaling factor(δ):** For the recursive relationship we need to intialize $S_D(-1)$ but to make sure that it doesn't affect the values of R_D we initialize it as δI as it needs to be diagonal and we need large δ so that $R_D(-1) = \delta^{-1} I$ is small and has no effect on further values.

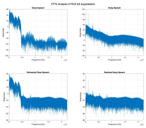
Notch filter: We used a cascaded 2nd order notch filter to preserve the tonal frequencies which is achieved by using the recurrence relation given by

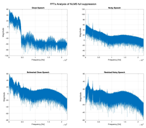
$$\boxed{y(n) = b_1 \cdot x(n) + b_2 \cdot x(n-1) + b_3 \cdot x(n-2) - a_1 \cdot y(n-1) - a_2 \cdot y(n-2)}$$

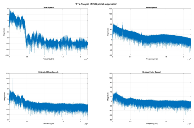
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

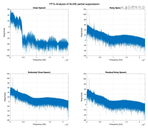
Observations

Results:









- In the case of full suppression for both cases by observing the estimated clean speech, we can say the noise components were suppressed significantly, but we can observe higher suppression in case of RLS.
- Similarly for partial suppression, we can observe RLS suppresses non-tonal noise components more effectively compared to the NLMS, As we can observe the frequency magnitudes to be lesser in the estimated clean speech graph. I have chosen the tonal component to be equal to 1000Hz for RLS and NLMS, we can observe it is being preserved from the residual noise graph.

Method comparisons:

- **RLS:** Useful in fast changing systems, more accurate estimation, more complex procedure , higher time complexity, more prone to numerical instability, requires higher memory,
- **NLMS:** lesser accuarcy as compared to RLS, lesser complex procedure , lesser time complexity, less prone to numerical instability, requires lower memory, useful only in steady systems

Design vulnerabilities

- Choice of lambda significantly affects the output. We need an idea of stationarity of the input process to choose suitable lambda.
- The time complexity is higher for this design and it is sensitive to initialising values and illconditioning of R_D

Final Observations:

- We observe that with $N = 12$, $\lambda = 0.99999$, $\delta = 10^5$, we observe SNR improvement of 39.15dB appx.
- Full suppression maximizes noise removal but risks slight speech distortion. From the plot we can observe that the speech components in the signal are almost conserved while the noise components are significantly suppressed.
- Partial suppression had preserved the tonal frequencies and attenuated the others significantly but due to the bandwidth of the notch there is slight distortion of other frequencies as well and also the phase response of the notch causes some distortion as well.

Assumption

- The microphones of the headphones have a built in ADC block.

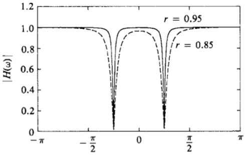
References and derivations

References

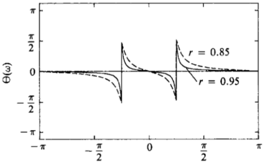
- M. H. Hayes(541-546), "Statistical Digital Signal Processing and Modeling", John Wiley Sons, 1996.
- Paulo S. R. Diniz, "Adaptive Filtering: Algorithms and Practical Implementation"- 5.1 and 5.2 (pg 157 - 160).
- John G. Proakis (4.4), Dimitris K. Manolakis, "Digital Signal Processing 4e"(pg 347 - 349).

Notch Filter

- $z_{1,2} = e^{\pm j\omega_0}$, $p_{1,2} = re^{\pm j\omega_0}$
  $H(z) = b_0 \dfrac{(1-e^{j\omega_0} z^{-1})(1-e^{-j\omega_0} z^{-1})}{(1-re^{j\omega_0} z^{-1})(1-re^{-j\omega_0} z^{-1})}$; b_0=1 because, the frequency components which were not notched should retain their values
  As H(z)=$\dfrac{Y(z)}{X(z)}$
- y(n)=$x(n)$ - 2 $\cos\omega_o x(n-1)$+x(n-2)+2$r\cos\omega_o y(n-1)$-$r^2 y(n-2)$
- $a_1 = -2r\cos\omega_o$, $a_2 = r^2$, $b_1 = 1$, $b_2 = -2\cos\omega_o$, $b_3 = 1$,(when compared with the coefficients of recurrence relation in notch filter in slide 3)
- As we want to have the bandwidth to be as small as possible we use a second order filter so that the poles can blow up the values around the zeroes in the magnitude response of the filter.



Magnitude Response



Phase Response