Adaptive Noise Cancellation in Headphones

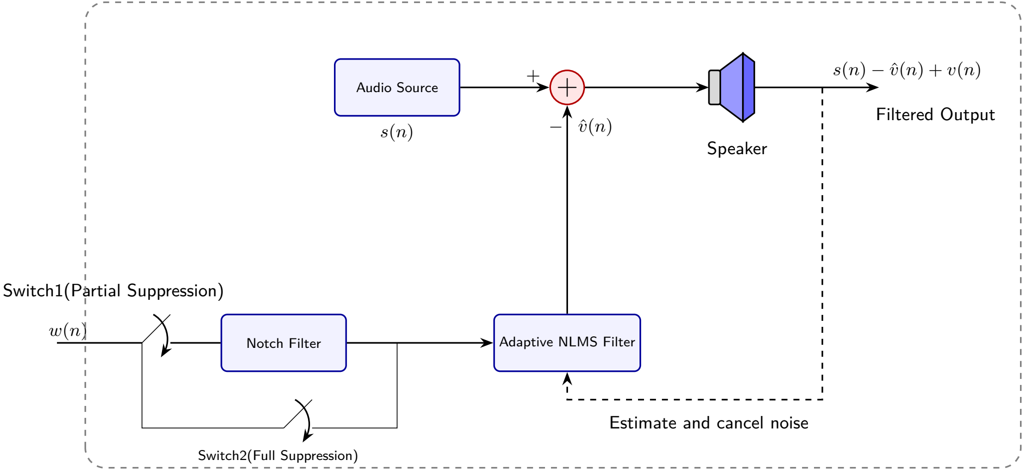
Full and Partial Noise Suppression using NLMS and Tonal Filtering

Kamale Goutham - *EE23BTECH11028*
Kanishk Devatwal - *EE23BTECH11029*
Kantyapalam Shravya - *EE23BTECH11030*
K Meghana Sai - *EE23BTECH11031*

EE2800 - Digital Signal Processing

April 2025

Block Diagram



Design Choices and Justifications

Full Suppression: NLMS filter estimates noise $\hat{v}(n)$ from reference $w(n)$ and subtracts it from noisy speech $d(n)$. Chosen for simplicity, low computational cost, and adaptive capability.

Partial Suppression: We add a notch filter to remove dominant tonal noise (e.g., 1000 Hz) from $w(n)$ and feeding into NLMS filter which gives estimated noise $\hat{v}(n)$ which doesn't have tonal components of $1000Hz$.

Why NLMS? In LMS, a large input can cause large weight updates, leading to instability and algorithm divergence, which reduces SNR. NLMS overcomes this by normalizing the step size with input power, ensuring more stable updates even with input fluctuations.

Parameters:

- Full Suppression: Filter order: $M = 8$, Step size: $\mu = 0.008$.
- Partial Suppression: Filter order: $M = 12$, Step size: $\mu = 0.00008$, Pole radius: $r = 0.998$.

Detailed Design, Pros and Cons

Pros:

- Around $26.56dB$ of SNR in full suppression.
- Programmable tonal notch frequency in partial suppression.
- Low latency feasible due to buffer-based NLMS unlike more complex algorithms like RLS.

Cons:

- Assumes ideal noise measurement $w(n)$.
- Step size sensitivity can affect performance.
- Fixed filter order limits flexibility.

Assumptions:

- The system must process each incoming sample within a time frame that does not cause noticeable delay.
- We assume that the external noise has strong Tonal components at the given frequency.

Notch Filter Derivation and References

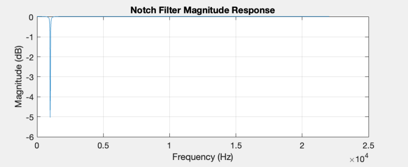
Notch Filter Transfer Function:

$$H(z) = \frac{(1 - e^{j\omega} z^{-1})(1 - e^{-j\omega} z^{-1})}{(1 - re^{j\omega} z^{-1})(1 - re^{-j\omega} z^{-1})}$$

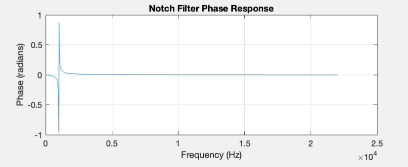
where $\omega = \frac{2\pi f_0}{F_s}$, and r controls the bandwidth of the notch.

Difference Equation: $y[n] = x[n] - 2\cos(\omega)x[n-1] + x[n-2] + 2r\cos(\omega)y[n-1] - r^2 y[n-2]$

Zeros: $z_{1,2} = e^{\pm j\omega}$ **Poles:** $p_{1,2} = re^{\pm j\omega}$



Magnitude Response



Phase Response

References:

- M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley, 1996.
- Paulo S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, Springer, 2020.