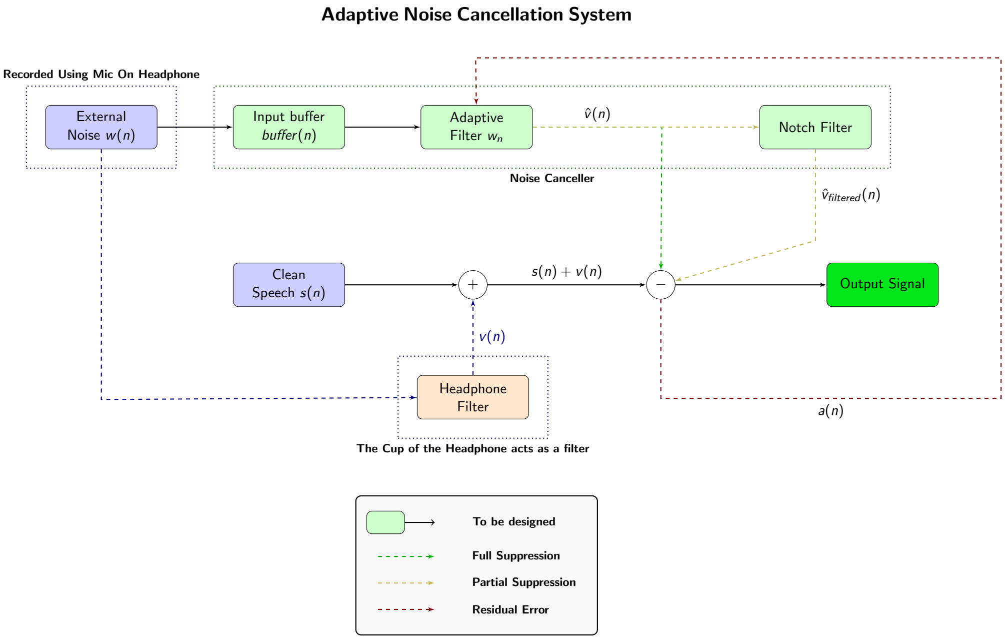
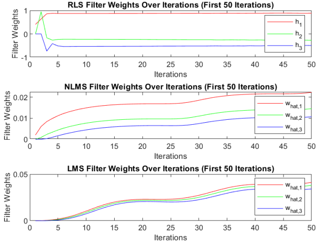
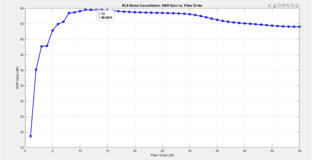
Block Diagram

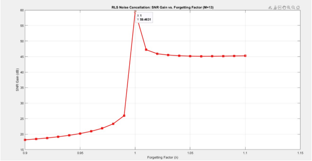


Design Choices

- We chose to use RLS rather than LMS or NLMS because RLS converges faster than LMS and NLMS.
- The RLS algorithm is more complex than LMS and NLMS, but it provides better performance in terms of convergence speed and steady-state error. We can see that from the plot filter coefficients v/s iterations
- The optimal value of the filter order p is determined by SNR v/s filter order plot for lambda = 1. The optimal value of p obtained is 13.
- The optimal value of the forgetting factor lambda is determined by SNR v/s forgetting factor plot for p = 13. The optimal value of lambda obtained is 1.







Noise Canceller Design

- The RLS algorithm follows the following Error Function:

$$E(n) = \sum_{i=0}^{n} \lambda^{n-i} e^2(i)$$
$$e(i) = d(i) - w_n^T x(i)$$

- After taking the derivative w.r.t w_n and setting it to zero and simplifying it, we get the following solution for the weights:

$$R_x(n) w_n = r_{dx}(n)$$

- We now try to calculate w_{n-1} using the above equation by calculating $R_x^{-1}(n)(P(n))$. We finally get the update equation for the weights as:

$$w_n = w_{n-1} + g(n)\alpha(n)$$

- Here,

$$g(n) = \frac{P(n)x(n)}{\lambda + x^T(n)P(n)x(n)}$$
$$\alpha(n) = e(n) - w_{n-1}^T x(n)$$

We can now iteratively update the weights and calculate the value of $\hat{v}(n)$ from the weights.

- For partial suppression, we filter the estimated noise $\hat{v}(n)$ during every iteration of weight update and noise estimation. The filter is a notch filter with the following transfer function:

$$H(z) = \frac{1 - 2\cos(\omega_0)z^{-1} + z^{-2}}{1 - 2r\cos(\omega_0)z^{-1} + rz^{-2}}$$
$$\implies \frac{Y(z)}{X(z)} = \frac{1 - 2\cos(\omega_0)z^{-1} + z^{-2}}{1 - 2r\cos(\omega_0)z^{-1} + rz^{-2}}$$
$$\implies y(n) = a_1 y(n-1) - a_2 y(n-2) + b_0 x(n) - b_1 x(n-1) + b_2 x(n-2)$$

References