# Solution Overview

**Adaptive Filter Used: Recursive Least Squares (RLS)**

- RLS is an adaptive filter that minimizes mean square error using an exponentially weighted least squares criterion.
- Well-suited for noise cancellation in dynamic environments due to its fast convergence and precision.

| Property | Advantage | Disadvantage |
|---|---|---|
| Convergence | Fast, near-instantaneous adaptation | Computationally expensive |
| Tracking Ability | Performs well in time-varying noise | High sensitivity to numerical errors |
| Memory Requirements | Uses inverse correlation matrix for updates | Consumes more memory than LMS |

**Processing Pipeline & Implementation Details**

- The cost function:

$$\boldsymbol{J(n) = \sum_{k=0}^{n} \lambda^{n-k} e^2(k)}$$

Where e(n) = estimated true signal and λ is the *forgetting factor.*

- The weights are updated every 32,768 samples.

## Update Algorithm

- $\boldsymbol{x(n), d(n), w(n)}$ are $n^{th}$ sequence component of noisy signal, true signal and weights of the filter resp.
- $\boldsymbol{P(k)} = \boldsymbol{R_{xx}^{-1}(k)}$ is the inverse-correlation matrix (to be initialized)
- Define $\boldsymbol{X(n)}$ as the column vector containing the last buffered samples of $\boldsymbol{x(n)}$.
- The algorithm is then as follows:

$$\text{Gain vector: } g(n) = \frac{P(n-1)X(n)}{\lambda + X^T(n)P(n-1)X(n)}$$

$$\text{Noise Estimate: } \hat{v}(n) = X^T(n)w(n-1)$$

$$\text{True Signal Estimate: } e(n) = d(n) - \hat{v}(n)$$

$$\text{Weight Update: } w(n) = w(n-1) + g(n)e(n)$$

$$\text{Inverse-Correlation Matrix Update: } \frac{1}{\lambda}\left(P(n-1) - g(n)X^T(n)P(n-1)\right)$$

## Computational Complexity (per iteration)

| Step | Complexity |
|---|---|
| Gain Vector | $O(N^2)$ |
| Noise Estimate | $O(N)$ |
| True Signal Estimate | $O(1)$ |
| Weight Update | $O(N)$ |
| Inverse-Correlation Matrix Update | $O(N^2)$ |

**Results**

Performance Metrics for Full Suppression:

Pre-Cancellation SNR: -16.77 dB

Post-Cancellation SNR: 33.19 dB

**Design Trade-offs & Justifications**

| Parameter | Choice | Justification |
|---|---|---|
| Update Rate (number of samples) | 32768 | Best performance (highest SNR for *full_suppression*) |
| Filter Order | 16 | Balances latency vs. performance |
| Forgetting Factor | 0.99999 | Balance between filter performance vs. no. of past errors |
| Initialization $P(0)$ | $\frac{I}{0.001}$ | Prevents numerical instability |

**Partial Suppression Approach under Consideration**

- Continuously apply a notch filter across the *external_noise* that removes the tonal frequency from reference.
- The algorithm will thus never learn to adapt to the tone and it will be retained in the output.

**References**

1. Paulo S. R. Diniz, ”Adaptive Filtering: Algorithms and Practical Implementation” (Chapter 5)

2. Ali, F., Rawat, P., & Malvia, S. (2017). Comparative analysis and survey of LMS and RLS adaptive algorithms. International Journal of Computer Applications, 161(3), 26–29. https://doi.org/10.5120/ijca2017913136

3. S.Haykin, "Adaptive Filter Theory", 3rd Edition, Prentice Hall, N.J., 1996 (Chapter 13)

4. M. H. Hayes, ”Statistical Digital Signal Processing and Modeling”, John Wiley & Sons, 1996 (Chapter 9)