Adaptive Noise Cancellation Using LMS, NLMS, and RLS Algorithms

AVD611 - Modern Signal Processing

Project Report

Submitted by:
Saurabh Kumar
SC22B146

Submitted to:
Dr. Chris Prema

Department of Avionics
Indian Institute of Space Science and Technology

Abstract

Noise is an unavoidable component in real-world signal acquisition systems and often degrades the quality and intelligibility of the desired signal. Adaptive Noise Cancellation (ANC) is a powerful technique that exploits the availability of a reference noise signal correlated with the noise contaminating the primary signal. Unlike fixed filters, adaptive filters automatically adjust their coefficients to minimize the mean square error between the filter output and a desired response. This project presents a detailed study of adaptive noise cancellation using three widely used adaptive algorithms: Least Mean Squares (LMS), Normalized Least Mean Squares (NLMS), and Recursive Least Squares (RLS). The theoretical foundations of adaptive filtering are discussed, followed by the derivation of the update equations for each algorithm. MATLAB simulations are performed to compare their convergence behavior and noise suppression capability. The results demonstrate the trade-off between convergence speed, computational complexity, and steady-state performance among LMS, NLMS, and RLS algorithms.

1 Problem Statement

In many practical scenarios, the desired signal of interest is corrupted by additive noise. The observed signal can be modeled as

$$d(n) = s(n) + v_0(n) \tag{1}$$

where $s(n)$ is the desired signal and $v_0(n)$ is an undesired noise component. In several applications, such as speech enhancement, biomedical signal processing, and communication systems, direct filtering of $d(n)$ is ineffective because the spectral characteristics of $s(n)$ and $v_0(n)$ may overlap.

Adaptive noise cancellation assumes the availability of an additional reference noise signal $v_1(n)$ that is correlated with $v_0(n)$ but uncorrelated with $s(n)$. The objective is to use an adaptive filter to estimate the noise component from the reference signal and subtract it from the primary signal, thereby recovering an estimate of the desired signal.

2 State of the Art

Adaptive filtering has been extensively studied since the pioneering work of Widrow and Hoff, who introduced the LMS algorithm. LMS is popular due to its simplicity and low computational complexity, making it suitable for real-time applications. However, its convergence speed is limited and highly dependent on the input signal statistics.

To address these limitations, the NLMS algorithm was developed, which normalizes the step size by the instantaneous input power. This modification improves stability and convergence robustness, especially when the input signal power varies with time. For applications requiring very fast convergence, the RLS algorithm is used. RLS achieves near-optimal performance by recursively computing the least-squares solution using all past data with exponential weighting. Although RLS offers superior convergence properties, it comes at the cost of increased computational complexity and numerical sensitivity.

3 Challenges in Adaptive Noise Cancellation

The effectiveness of adaptive noise cancellation depends on several practical challenges. First, the reference noise must be sufficiently correlated with the noise present in the primary signal; otherwise, noise estimation becomes impossible. Second, the adaptive filter must balance convergence speed and steady-state error through appropriate parameter selection. Third, computational complexity and numerical stability are important considerations, especially for high-order filters and real-time implementations. Finally, nonstationary environments require the algorithm to track changes in noise characteristics without sacrificing stability.

4 Adaptive Noise Cancellation Approach

The adaptive noise cancellation system consists of an FIR adaptive filter driven by the reference noise signal $v_1(n)$. The filter output

$$y(n) = \mathbf{h}^T(n)\mathbf{x}(n) \tag{2}$$

is an estimate of the noise component $v_0(n)$, where

$$\mathbf{x}(n) = [v_1(n), v_1(n-1), \ldots, v_1(n-M+1)]^T$$

and $\mathbf{h}(n)$ is the adaptive filter coefficient vector.

The error signal is defined as

$$e(n) = d(n) - y(n) \tag{3}$$

which serves as both the system output and the adaptation criterion. The adaptive filter coefficients are updated to minimize the mean square error $E\{e^2(n)\}$.

5 Least Mean Squares (LMS) Algorithm

The LMS algorithm is derived using a stochastic gradient descent approach to minimize the cost function

$$J = E\{e^2(n)\}.$$

The weight update equation is given by

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu\,\mathbf{x}(n)e(n) \tag{4}$$

where μ is the step size controlling convergence speed and stability. LMS is computationally efficient but exhibits slow convergence when the input signal is highly correlated.

6 Normalized Least Mean Squares (NLMS) Algorithm

The NLMS algorithm improves upon LMS by normalizing the step size with respect to the input signal power. The update equation is

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\mu}{\mathbf{x}^T(n)\mathbf{x}(n) + \gamma}\mathbf{x}(n)e(n) \tag{5}$$

where γ is a small positive constant to avoid division by zero. NLMS provides improved convergence behavior and robustness to input power variations.

7 Recursive Least Squares (RLS) Algorithm

The RLS algorithm minimizes an exponentially weighted least squares cost function:

$$J(n) = \sum_{i=0}^{n} \lambda^{n-i} e^2(i) \tag{6}$$

where λ is the forgetting factor. The RLS update equations are

$$\mathbf{g}(n) = \frac{\mathbf{P}(n-1)\mathbf{x}(n)}{\lambda + \mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)} \tag{7}$$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{g}(n)e(n) \tag{8}$$

$$\mathbf{P}(n) = \frac{1}{\lambda}\left[\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\right] \tag{9}$$

RLS offers very fast convergence and excellent tracking performance at the cost of higher computational complexity.

8 Why Adaptive Noise Cancellation Works

Adaptive noise cancellation works due to the orthogonality principle. At convergence, the error signal becomes orthogonal to the reference input:

$$E\{\mathbf{x}(n)e(n)\} = 0.$$

Since the desired signal $s(n)$ is uncorrelated with the reference noise $v_1(n)$, the adaptive filter models only the noise component. Consequently, the error signal converges to an estimate of the desired signal.

9 Simulation Results

MATLAB simulations were performed to evaluate the performance of LMS, NLMS, and RLS algorithms. The results demonstrate differences in convergence speed, steady-state error, and robustness.

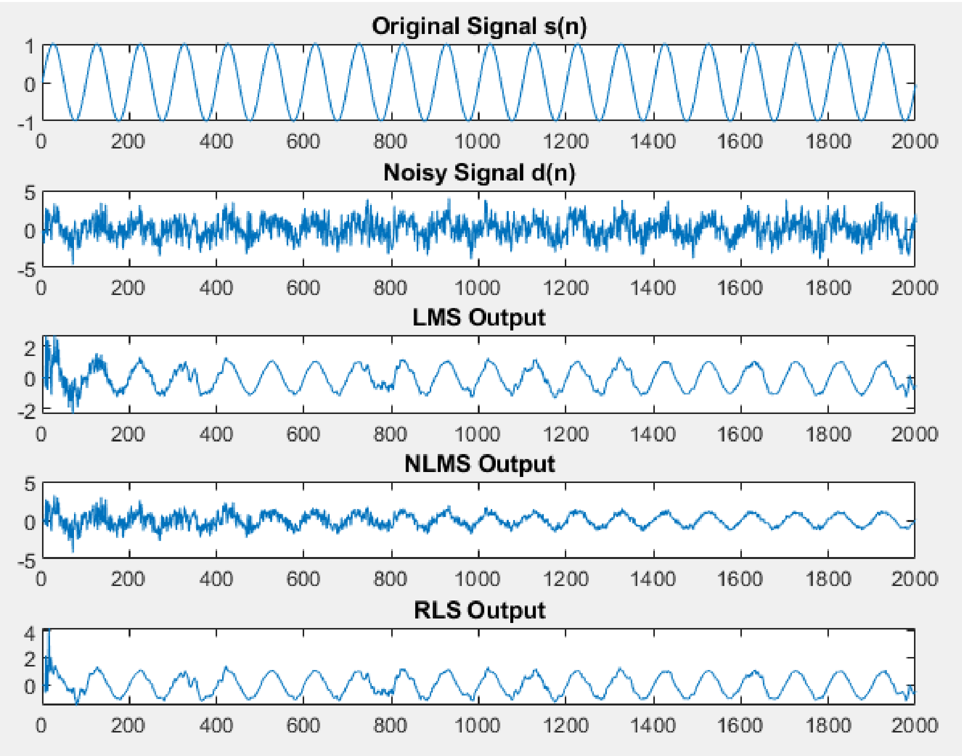

Figure 1: Recovered signal using LMS, NLMS, and RLS algorithms

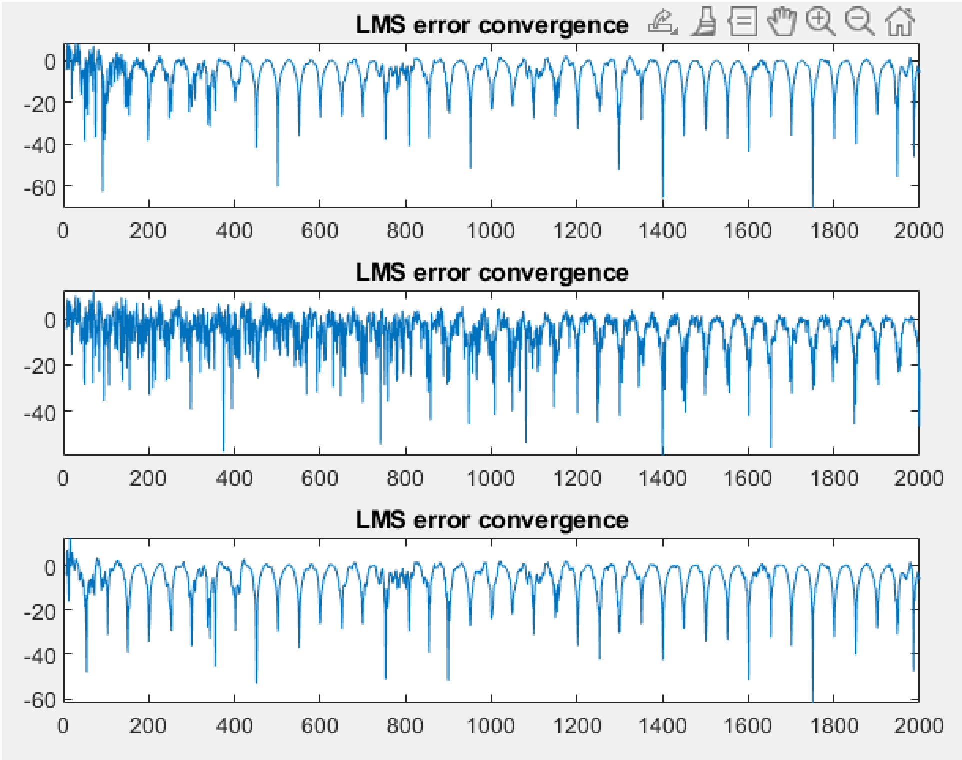

Figure 2: Error convergence comparison in dB

10 Conclusion

This project presented a comprehensive study of adaptive noise cancellation using LMS, NLMS, and RLS algorithms. LMS offers simplicity and stability but slow convergence, NLMS improves robustness through normalization, and RLS provides rapid convergence and superior tracking performance. The choice of algorithm depends on application requirements such as computational resources, convergence speed, and environmental nonstationarity. Adaptive noise cancellation remains a powerful and widely applicable technique in modern signal processing systems.

A MATLAB Code

Appendix A: MATLAB implementation of LMS, NLMS, and RLS algorithms used in this project.

```
%% Parameters
N = 2000; % Number of samples
n = (0:N-1)';

%% Desired signal
```

```
s = sin(2*pi*0.01*n);

%% Noise generation
v1 = randn(N,1); % Reference noise
v0 = filter([1 0.5],1,v1); % Correlated noise

%% Primary signal
d = s + v0;

%% Filter length
M = 8;

%% LMS
step_lms = 0.01;
h_lms = zeros(M,1);
y_lms = zeros(N,1);
e_lms = zeros(N,1);

for k = M:N
    x = v1(k:-1:k-M+1);
    y_lms(k) = h_lms.' * x;
    e_lms(k) = d(k) - y_lms(k);
    h_lms = h_lms + step_lms * x * e_lms(k);
end

%% NLMS
step_nlms = 0.01;
gamma = 1e-6;
h_nlms = zeros(M,1);
y_nlms = zeros(N,1);
e_nlms = zeros(N,1);

for k = M:N
    x = v1(k:-1:k-M+1);
    y_nlms(k) = h_nlms.' * x;
    e_nlms(k) = d(k) - y_nlms(k);
    h_nlms = h_nlms + (step_nlms/(x.'*x+gamma)) * x * e_nlms(k);
end
```

```
%% RLS
lambda = 0.99;
delta = 0.1;
h_rls = zeros(M,1);
P = (1/delta)*eye(M);
y_rls = zeros(N,1);
e_rls = zeros(N,1);

for k = M:N
    x = v1(k:-1:k-M+1);
    y_rls(k) = h_rls.' * x;
    e_rls(k) = d(k) - y_rls(k);
    g = (P*x)/(lambda + x.'*P*x);
    h_rls = h_rls + g*e_rls(k);
    P = (P - g*x.'*P)/lambda;
end

%% Plots
figure;
subplot(5,1,1);
plot(s); title('Original Signal s(n)');

subplot(5,1,2);
plot(d); title('Noisy Signal d(n)');

subplot(5,1,3);
plot(e_lms); title('LMS Output');

subplot(5,1,4);
plot(e_nlms); title('NLMS Output');

subplot(5,1,5);
plot(e_rls); title('RLS Output');

%% Error convergence comparison
figure;
subplot(3,1,1)
plot(10*log10(e_lms.^2)); title('LMS error convergence');
```

```
subplot(3,1,2)
plot(10*log10(e_nlms.^2)); title('LMS error convergence');

subplot(3,1,3)
plot(10*log10(e_rls.^2)); title('LMS error convergence');
```