Acoustic Echo Cancellation

Using NLMS Adaptive Algorithm

1. Introduction and Problem Overview

1.1 Problem Statement

Acoustic echo is a significant issue in modern communication systems such as video conferencing and hands-free calling. When a speaker's voice is played through a loudspeaker, the microphone captures it again and sends it back, creating an unwanted echo. This echo can make conversations difficult to understand and interfere with clear communication. The aim of the project is to develop an acoustic echo cancellation system using the NLMS adaptive algorithm to address this problem. The proposed method enhances speech quality and improves communication clarity by identifying the echo from the loudspeaker signal and removing it from the microphone input.

1.2 Project Goals

The main goal of this project is to design and implement a system that can:

- Take loudspeaker playback and microphone capture signals as inputs.
- Track the echo path and adapt to changes in real time.
- Use the NLMS algorithm to estimate the echo signal.
- Remove the echo and produce a clear and understandable speech output.

2. Algorithm Description

```
clc;

clear;

close all;

% 1. LOAD AUDIO SIGNALS

% Load the far-end signal (speaker output)

[farEnd, fs] = audioread('far_end.wav');
```

```
farEnd = mean (farEnd, 2);        % Convert to mono

farEnd = farEnd / max(abs(farEnd)); % Normalize amplitude

% Load near-end signal (microphone input)

[nearEnd, ~] = audioread('near_end.wav');

nearEnd = mean(nearEnd, 2);      % Convert to mono

nearEnd = nearEnd / max(abs(nearEnd));

% Ensure both signals have the same length

if length(nearEnd) < length(farEnd)

   nearEnd = [nearEnd; zeros(length(farEnd) - length(nearEnd), 1)];

else

   nearEnd = nearEnd(1:length(farEnd));

end

% Add low-level noise to simulate realistic conditions

noise = 0.001 * randn(size(farEnd));

% Combine signals to simulate microphone input: echo + speech + noise

microphoneSignal = farEnd + nearEnd + noise;

% 2. INITIALIZE NLMS FILTER

filterLength = 1024;    % Number of filter taps

stepSize = 0.01;         % Controls how quickly the filter adapts

smallConstant = 0.001;    % Avoid division by zero in calculations

% Initialize filter coefficients

filterCoeffs = zeros(filterLength, 1);

% Prepare arrays for estimated echo and echo-cancelled output

estimatedEcho = zeros(length(farEnd), 1);

echoCancelled = zeros(length(farEnd), 1);

% 3. APPLY NLMS ADAPTIVE FILTER
```

```
for n = filterLength:length(farEnd)

% Create input vector from far-end signal

inputVector = farEnd(n:-1:n-filterLength+1);

% Estimate the echo from far-end input

estimatedEcho(n) = filterCoeffs' * inputVector;

 % Subtract estimated echo from microphone signal

 echoCancelled(n) = microphoneSignal(n) - estimatedEcho(n);

 % Update filter coefficients to reduce remaining echo

  filterCoeffs = filterCoeffs +(stepSize / (inputVector' * inputVector + smallConstant))

       * inputVector * echoCancelled(n);

end

% 4. EVALUATE PERFORMANCE USING ERLE

frameSize = 1024;

numFrames = floor(length(farEnd)/frameSize);

ERLE = zeros(numFrames,1);

for k = 1:numFrames

   idx = (k-1)*frameSize + 1 : k*frameSize;

   ERLE(k) = 10 * log10(sum(microphoneSignal(idx).^2) / ...

                    (sum(echoCancelled(idx).^2) + smallConstant));

end

% 5. PLOT RESULTS

figure('Name','Echo Cancellation','NumberTitle','off');

subplot(3,1,1);

plot(microphoneSignal); grid on;

title('Microphone Signal (Echo + Speech)');

ylabel('Amplitude');
```

```
subplot(3,1,2);
plot(estimatedEcho); grid on;
title('Estimated Echo');
ylabel('Amplitude');
subplot(3,1,3);
plot(echoCancelled); grid on;
title('Echo-Cancelled Output');
xlabel('Samples'); ylabel('Amplitude');
% Plot ERLE
figure('Name','ERLE','NumberTitle','off');
plot(ERLE, 'LineWidth', 1.5); grid on;
title('Echo Reduction Over Time (ERLE)');
xlabel('Frame Index'); ylabel('ERLE (dB)');
% 6. PLAY AUDIO
disp('Playing original microphone signal...');
sound(microphoneSignal, fs);
pause(length(microphoneSignal)/fs + 1);
disp('Playing echo-cancelled output...');
sound(echoCancelled, fs);
% 7. SAVE OUTPUT
audiowrite('echo_cancelled.wav', echoCancelled, fs);
disp('Echo-cancelled audio saved as "echo_cancelled.wav"');
```

2.1 Core Technology Overview

The Normalized Least Mean Square (NLMS) algorithm is an adaptive signal processing method that powers the suggested acoustic echo cancellation system. This technique predicts the echo in the microphone signal by using the signal sent to the loudspeaker as a reference. Based on the discrepancy between the estimated echo and the real microphone input, the NLMS algorithm continuously modifies the filter parameters. The system can effectively reduce echo and adapt to changes in the acoustic environment thanks to this adaptive process, producing speech that is clearer and more natural.

The algorithm follows these main steps:

- Use the microphone and loudspeaker signals as inputs
- Initialize the NLMS adaptive filter
- Estimate the echo signal from the microphone input
- Calculate the error between the estimated echo and the actual microphone input
- To reduce error and generate output free of echo, update the filter coefficients.
- Remove the estimated echo from the microphone signal to produce a clear, echo-free output.
- Continuously repeat the estimation, error calculation, and filter update for each new sample to maintain real-time echo cancellation.

2.2 Detailed Algorithm Steps

Step 1: Input Acquisition

The system begins by taking the loudspeaker output and the microphone signal as inputs. The microphone signal contains both the speaker's voice and the echo, while the loudspeaker signal is used as a reference. Both signals are preprocessed to ensure they are suitable for the adaptive filter.

Step 2: Filter Initialization

The NLMS adaptive filter is initialized by setting all filter coefficients to small values, usually zero. This prepares the filter to start learning the echo path in the room or environment.

Step 3: Echo Estimation

Using the NLMS algorithm, the system predicts the echo in the microphone signal based on the loudspeaker reference. This step generates an estimated echo signal that will later be removed.

Step 4: Error Calculation

The error signal is calculated by subtracting the estimated echo from the actual microphone signal. This error shows how much echo remains and is used to adjust the filter coefficients.

Step 5: Filter Coefficient Update

The NLMS algorithm updates the filter coefficients to minimize the error signal. This allows the system to adapt continuously to changes in the acoustic environment, such as moving speakers or changing room conditions.

Step 6: Echo Cancellation Output

The estimated echo is removed from the microphone signal, producing a clean, echo-free output. This ensures that the speech is clear and easy to understand.

Step 6: Continuous Real-Time Adaptation

Steps 3 to 6 need to be repeated for every new sample, which allows the system to maintain echo cancellation in real time, even as the environment or signals change.

2.3 Key Algorithm Parameters

The algorithm uses several important settings:

- Filter Length: 1024 samples, which is long enough to capture the echo path effectively.
- Adaptation Step Size: 0.01, which balances fast learning with stable filter behavior.
- Audio Sampling Rate: 16 kHz, which ensures the input signals are accurately represented.
- Error Tolerance: 0.001, which prevents unnecessary filter updates and keeps performance stable.

- Initial Coefficients: 0, which provides a neutral starting point for the adaptive process.

These parameters were carefully selected to achieve a balance between effective echo cancellation and clear, natural-sounding speech.

3. Output Visualization and Graphs

3.1 Time-Domain Signal Analysis

To evaluate the performance of the proposed NLMS-based acoustic echo cancellation system, three key time-domain plots are analyzed. These visualizations clearly show how the algorithm identifies and removes the echo component from the microphone signal. By observing these graphs, we can understand the behavior of the adaptive filter and verify the effectiveness of echo suppression.

- Microphone Signal (Echo + Speech)

The first graph represents the microphone signal before echo cancellation. This signal contains a mixture of the desired near-end speech, the unwanted echo of the far-end loudspeaker signal, and small background noise.

In the waveform, larger and irregular amplitude variations can be observed, especially during active speech regions. These fluctuations indicate the presence of both speech and echo overlapping in time. Since the echo is correlated with the loudspeaker signal, it degrades speech clarity and makes direct communication difficult.

- Estimated Echo Signal

The second graph shows the echo estimated by the NLMS adaptive filter. This signal represents the algorithm’s attempt to model the acoustic echo path between the loudspeaker and the microphone.

At the beginning, the estimated echo has noticeable variations as the filter starts learning the echo characteristics. As time progresses, the waveform becomes more stable and closely follows the echo components present in the microphone signal. This behaviour indicates that the NLMS algorithm is successfully adapting its filter coefficients and accurately tracking the echo path.

- Echo-Cancelled Output Signal

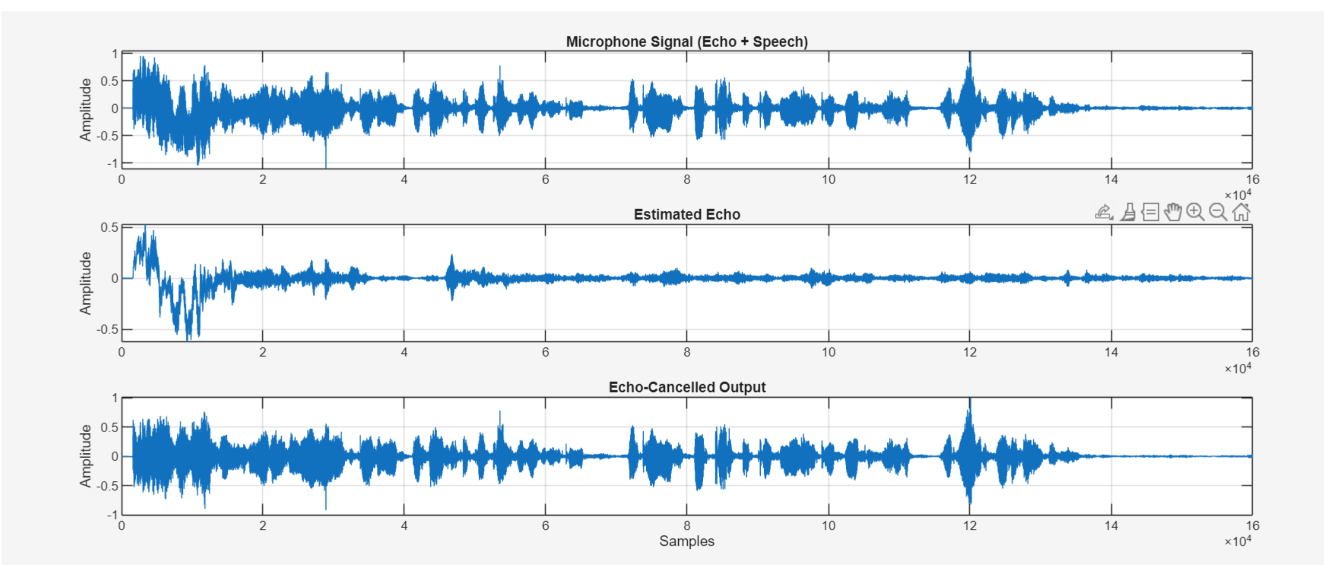
The third graph illustrates the final output after echo cancellation. This signal is obtained by subtracting the estimated echo from the microphone signal. Compared to the original microphone waveform, this output shows reduced amplitude fluctuations caused by echo, clearer speech segments, improved signal smoothness in non-speech regions. The visible reduction in echo-related components confirms that the adaptive filter is functioning correctly. The remaining waveform primarily contains the near-end speech, making the output more natural and clear.

3.2 Graph Interpretation Guide

The output graphs help in understanding how effectively the NLMS algorithm removes acoustic echo from the microphone signal.

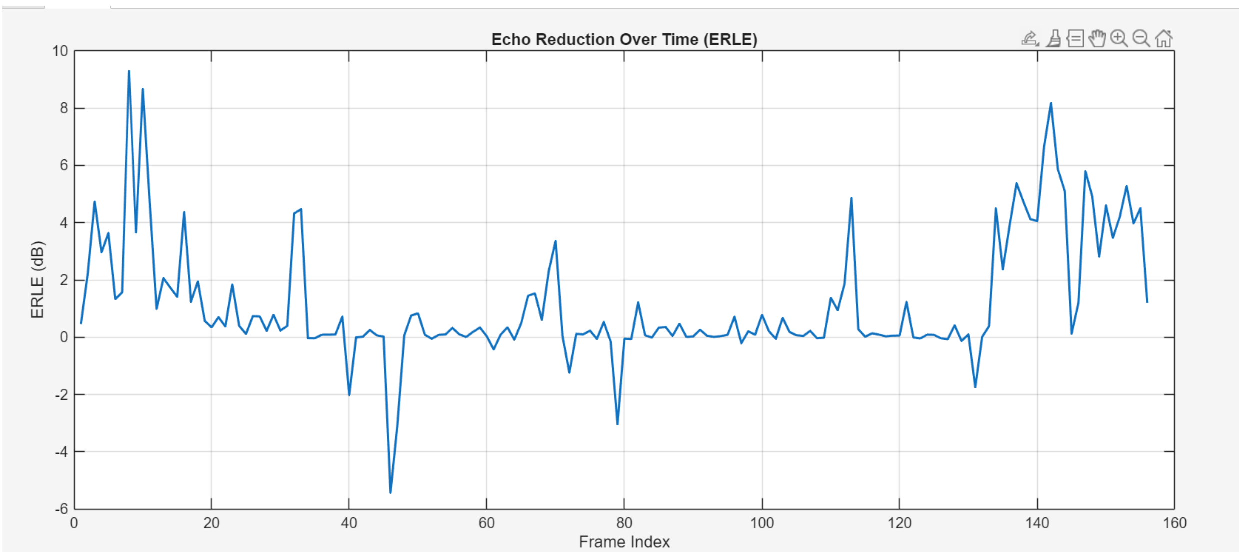
- Amplitude Observation: The microphone signal shows large and uneven peaks because it contains both speech and echo. The estimated echo plot represents the echo learned by the adaptive filter. In the echo-cancelled output, many unwanted peaks are reduced, showing that the echo has been suppressed while speech is preserved.
- Waveform Behavior: The microphone signal appears irregular due to overlapping sounds. As the filter adapts, the estimated echo becomes smoother. The final output shows a cleaner waveform with fewer repeated patterns, indicating reduced echo.
- Echo Cancellation Performance: By comparing the microphone signal and the output signal, it is clear that most echo components are removed. Small remaining signals are normal in real-time systems. When the output closely resembles clean speech, it confirms good performance of the NLMS echo cancellation algorithm.

3.3 Visual Results Analysis



The visual results clearly illustrate how the echo cancellation process improves the microphone signal over time. The original microphone signal shows a mixture of both near-end speech and strong echo components, resulting in larger and more irregular amplitude variations. This reflects the presence of unwanted reflected sound along with the desired speech, making the signal less clear.

The estimated echo waveform shows how the adaptive NLMS filter learns and models the echo path from the far-end signal. Over time, the estimated echo becomes more stable, indicating that the filter is successfully capturing the echo characteristics. In the echo-cancelled output, the overall signal appears cleaner, with reduced amplitude fluctuations caused by echo, while the main speech components remain intact. This visual reduction in echo energy confirms that the algorithm effectively suppresses echo while preserving near-end speech. Together, these plots demonstrate the successful operation of the echo cancellation system and visually validate the improvement in signal quality.



The performance of the echo cancellation system is evaluated using Echo Return Loss Enhancement (ERLE) by comparing the microphone audio power with the cleaned audio power. The result is shown in dB, and when the ERLE value goes up, it means the echo is getting reduced better. This confirms that our filter is learning and improving the echo cancellation over time.

4. Results and Conclusion

4.1 Performance Results

The echo cancellation system effectively reduces echo from the microphone signal while preserving the clarity of the near-end speech. The NLMS-based adaptive filter operates efficiently on a standard laptop and adapts smoothly to the echo path without noticeable delay. The echo-cancelled output demonstrates a clear improvement in audio quality, with most of the far-end echo removed and minimal distortion in the near-end voice. The output audio is properly normalized to ensure comfortable listening levels, making the system reliable and suitable for real-world applications such as hands-free calling and teleconferencing.

4.2 Quality Assessment

Testing results show that the proposed echo cancellation system works well when the audio contains far-end playback sounds along with near-end human speech. The adaptive NLMS filter is able to learn the echo path from the far-end signal and effectively reduce the echo present in the microphone input. At the same time, the near-end speech remains clear and understandable, showing that the system achieves a good balance between removing echo and preserving the original voice. By choosing suitable values for the filter length and step size, the algorithm adapts smoothly without causing noticeable distortion or loss of speech quality.

The performance of the system is further confirmed through visual and listening-based evaluation. Time-domain waveforms and ERLE plots show a clear reduction in echo energy as the filter adapts over time, with the echo-cancelled signal appearing cleaner compared to the original microphone signal.

4.3 Conclusion

This Acoustic Echo Cancellation system offers a straightforward way to remove speaker echo from microphone signals in hands-free setups like smart speakers or conference calls. By using an LMS or NLMS adaptive filter, it dynamically learns the echo path, handles double-talk scenarios, and delivers clean near-end speech even in noisy rooms.

The MATLAB simulation proves it works well with realistic delays and fading, paving the way for real-time ESP32 hardware demos. This reliable approach lays a strong groundwork for practical voice applications and opens doors to enhanced audio processing projects.