Nirma University

Institute of Technology

Department of Electronics and Communication Engineering

2EC402CC23 Digital Signal Processing

SEM-4

PROJECT REPORT

Adaptive Noise Cancellation

Project By: 23BEC153 (RAHUL KUMAR AGARWAL)

23BEC144 (SNEHA PATEL)

PROBLEM DESCRIPTION

Problem Description: Adaptive Noise Cancellation System

- In today’s world, background noise is a major challenge in audio processing, affecting everything from phone calls to medical devices and voice assistants. Traditional noise reduction methods often struggle when the noise is unpredictable or changing over time.
- The goal of this project is to develop an Adaptive Noise Cancellation (ANC) system that can intelligently remove unwanted noise from an audio signal using adaptive filters. Unlike fixed filters, which work best for predictable noise, adaptive filters continuously adjust themselves based on the changing noise environment.
- To overcome this issue , we will implement the **Least Mean Squares (LMS) algorithm**, which continuously updates the filter coefficients to minimize noise while preserving the desired signal. The system will take:
- **Primary Input:** A combination of the desired audio signal (e.g., speech, music) and unwanted noise.
- **Reference Input:** A separate recording of the noise, which is correlated with the noise in the primary input.
- **Output:** A noise-reduced version of the primary input, where the unwanted noise is significantly suppressed.
- This approach is widely used in hearing aids, hands-free communication, biomedical signal processing, and active noise-cancelling headphones.

ALGORITHM

Step 1: Input Signals

- Take two input signals:
  - **Primary Signal (S)**: Contains both the desired signal and noise.
  - **Reference Noise (N)**: Contains only the noise, which is correlated with the noise in the primary signal.

Step 2: Initialize the Adaptive Filter

- Start with an adaptive filter (like LMS or RLS).
- Set initial filter weights to small random values.

Step 3: Process the Signals

- Pass the reference noise through the adaptive filter.
- The filter adjusts itself to create an estimate of the noise present in the primary signal.

Step 4: Subtract Noise Estimate

- Subtract the estimated noise from the primary signal.
- This gives an output signal that mostly contains the desired signal with reduced noise.

Step 5: Update the Filter (Learning Process)

- Measure the error (difference between desired signal and noise estimate).
- Use an algorithm (e.g., LMS) to adjust filter weights so that the noise estimate improves over time.

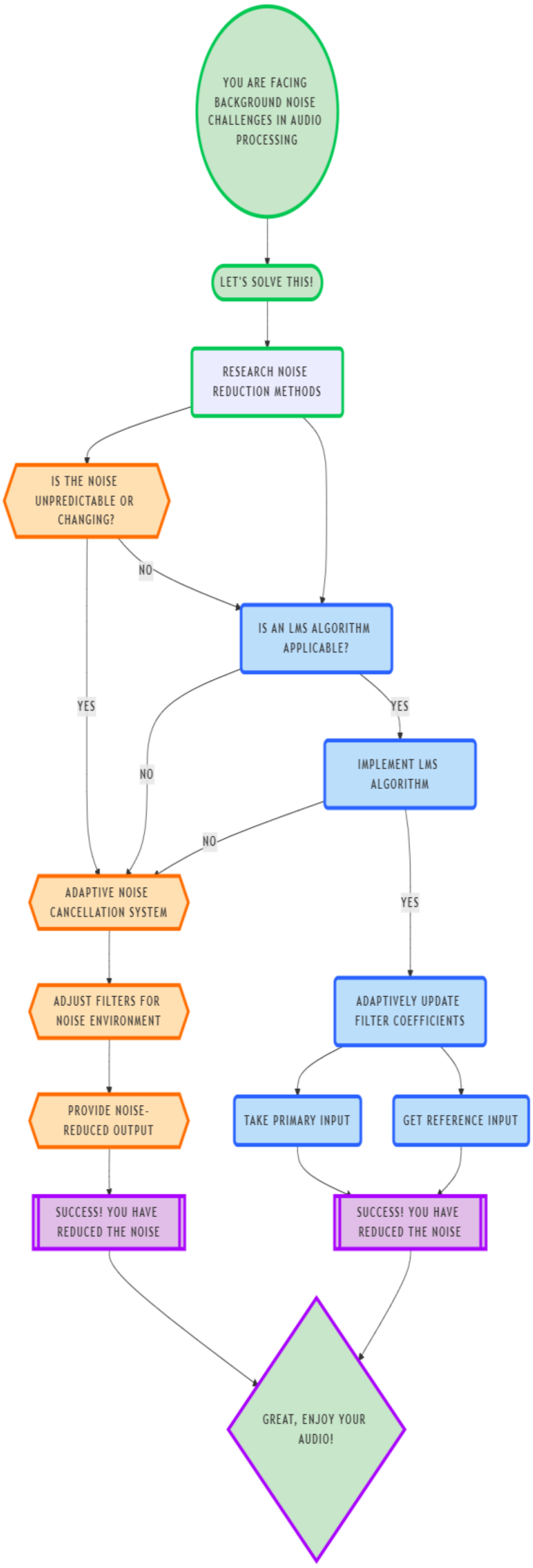
Step 6: Repeat Until Noise is Minimized

- Keep updating the filter and repeating the process.
- Stop when the noise is sufficiently canceled, or when the system reaches steady performance.

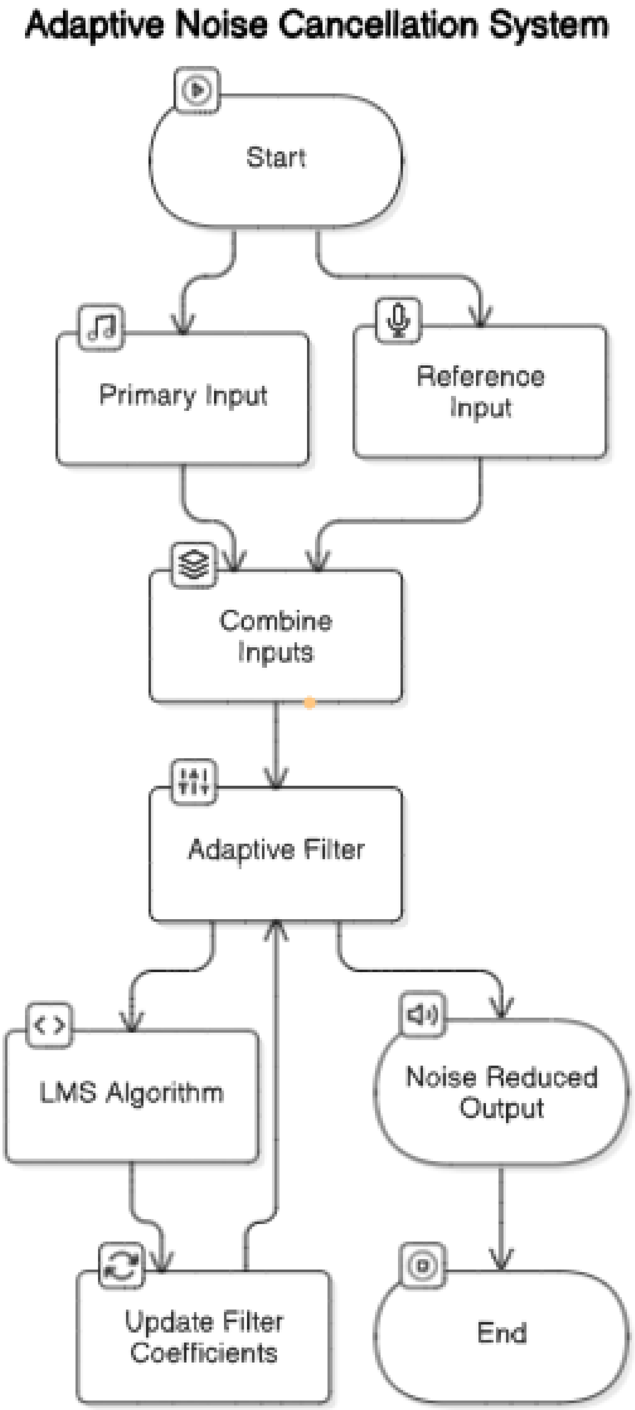
Step 7: Output the Clean Signal

- The final signal is sent out with minimal noise.

FLOWCHART


YOU ARE FACING BACKGROUND NOISE CHALLENGES IN AUDIO PROCESSING
LET'S SOLVE THIS!
RESEARCH NOISE REDUCTION METHODS
IS THE NOISE UNPREDICTABLE OR CHANGING?
NO
IS AN LMS ALGORITHM APPLICABLE?
YES
YES
NO
IMPLEMENT LMS ALGORITHM
NO
ADAPTIVE NOISE CANCELLATION SYSTEM
YES
ADJUST FILTERS FOR NOISE ENVIRONMENT
ADAPTIVELY UPDATE FILTER COEFFICIENTS
PROVIDE NOISE-REDUCED OUTPUT
TAKE PRIMARY INPUT
GET REFERENCE INPUT
SUCCESS! YOU HAVE REDUCED THE NOISE
SUCCESS! YOU HAVE REDUCED THE NOISE
GREAT, ENJOY YOUR AUDIO!

BLOCK DIAGRAM



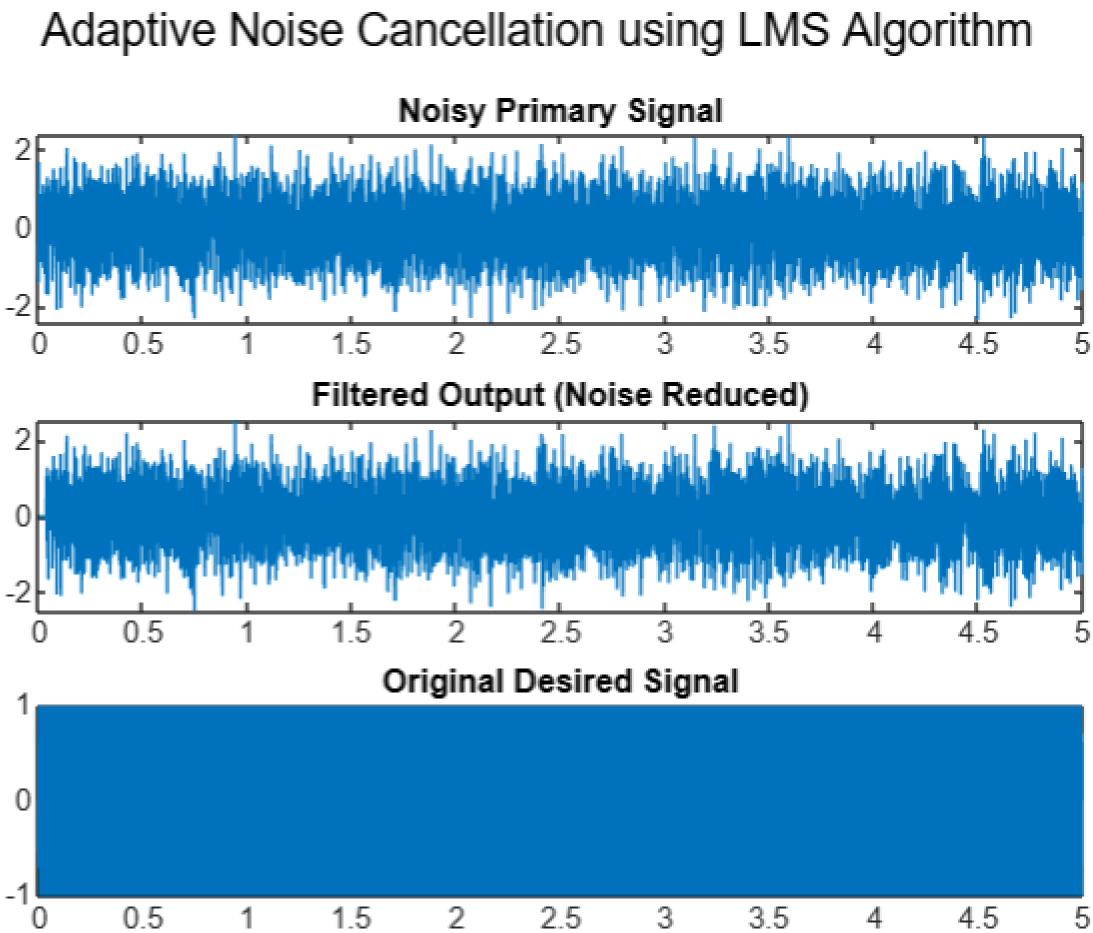
IMPLEMENTATION

(MATLAB CODE)

```
% User Inputs
fs = input('Enter sampling frequency (Hz): ')  % Sampling Frequency
duration = input('Enter signal duration (seconds): ');  % Duration of signal
freq = input('Enter frequency of desired signal (Hz): ');  % Desired signal frequency
noise_level = input('Enter noise amplitude (0-1): ');  % Noise level
M = input('Enter number of filter taps (M): ');  % Adaptive filter taps
mu = input('Enter learning rate (mu): ');  % Learning rate

% Time Vector
t = 0:1/fs:duration;

% Generate Signals
desired_signal = sin(2 * pi * freq * t);
noise = noise_level * randn(size(t));
primary_signal = desired_signal + noise;
reference_signal = [zeros(1, 10), noise(1:end-10)];

% LMS Adaptive Filtering
N = length(primary_signal);
W = zeros(M, 1);
error_signal = zeros(1, N);

for n = M:N
    x = reference_signal(n:-1:n-M+1)';
    y = W' * x;
    error_signal(n) = primary_signal(n) - y;
    W = W + mu * x * error_signal(n);
end

% Plot Results
figure;
subplot(3,1,1);
plot(t, primary_signal);
title('Noisy Primary Signal');
subplot(3,1,2);
plot(t, error_signal);
title('Filtered Output (Noise Reduced)');
subplot(3,1,3);
plot(t, desired_signal);
title('Original Desired Signal');
sgtitle('Adaptive Noise Cancellation using LMS Algorithm');
```

EXPERIMENT RESULT AND OBSERVATIONS

(OUTPUT WAVEFORMS)



CONCLUSION

After performing the **Adaptive Noise Cancellation System** using the **LMS Algorithm**, we successfully demonstrated the effectiveness of adaptive filtering in reducing unwanted noise from an audio signal. The results show that the LMS filter continuously adapts to the noise environment and minimizes the noise while preserving the desired signal.

Observations:

- The filtered output (error signal) has significantly reduced noise compared to the noisy primary signal.
- The effectiveness of noise cancellation depends on parameters such as filter length (M) and learning rate (mu).
- A higher number of filter taps (M) leads to better noise cancellation but increases computational complexity.
- A lower learning rate (mu) results in more stable adaptation but slower convergence, whereas a higher mu may cause instability.
- The algorithm performs well in real-time noise cancellation applications, making it useful for hearing aids, communication systems, and biomedical signal processing.

Limitations & Future Improvements:

- The performance may degrade if the reference signal is not well-correlated with the actual noise.
- Choosing an optimal learning rate is crucial—too high causes instability, too low slows adaptation.
- Future enhancements could include using Normalized LMS (NLMS) or Recursive Least Squares (RLS) algorithms for faster and more efficient adaptation.

REFERENCES

- MATLAB Documentation: https://www.mathworks.com/help/matlab/
- Advanced Digital Signal Processing by Shaila D. Apte