Line echo cancellation

Omar Masalmah: 1200060, Musab Masalmah: 1200078, Wasim Atta: 1200920

Abstract — The Line Echo Cancellation project aims to improve voice quality in communication systems by implementing an adaptive line echo canceller. Echoes caused by circuit mismatches interfere with the desired signal, degrading the overall quality. The project involves tasks such as analyzing impulse and frequency responses, evaluating the power spectrum density of a synthetic speech signal, estimating input and output powers, and implementing an adaptive filter using the normalized least mean squares algorithm. The performance of the adaptive filter is evaluated, and the estimated FIR channel is compared with the given FIR system. The project emphasizes teamwork, practical application of digital signal processing concepts, and problem-solving skills.

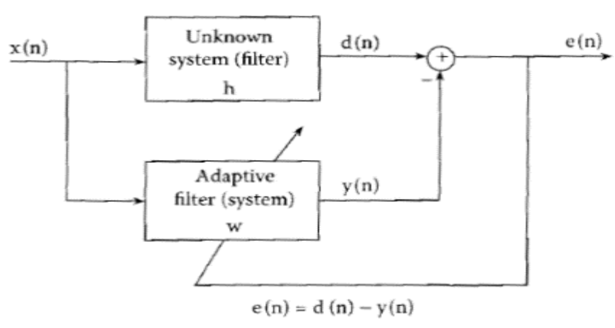
I. INTRODUCTION

The "Line Echo Cancellation" project focuses on mitigating echo interference in communication over phone lines. Teams of three students work together to implement an adaptive line echo canceller (LEC) to improve voice quality. The project involves analyzing the impulse and frequency responses of the echo path, plotting samples and spectra of a composite source signal, estimating input/output powers and echo-return-loss (ERL), training an adaptive filter with 128 taps, comparing different adaptive algorithms, and presenting findings in a mini-report. The project aims to enhance practical skills in digital signal processing and teamwork while addressing the challenge of echo cancellation in real-world communication systems.

II. PROBLEM SPECIFICATION

The problem addressed in the "Line Echo Cancellation" project is the interference of echo in communication over phone lines. When a signal travels from a far-end point to a near-end point, it is reflected at the near-end due to circuit mismatches, creating an attenuated replica of the signal known as an echo. This echo interferes with the quality of the received signal, impacting communication clarity. The project aims to develop and implement an adaptive line echo canceller (LEC) to mitigate the echo and improve voice quality at both ends of the communication. The LEC system is trained using input data from the far-end signal and reference data from the echo signal, allowing it to estimate and cancel out the echo effectively. The project also involves analyzing the impulse and frequency responses of the echo path, evaluating signal powers and echo-return-loss (ERL), comparing adaptive algorithms, and presenting the findings in a mini-report. By addressing the problem of echo interference, the project aims to enhance understanding and skills in digital signal processing and adaptive filtering techniques.

Figure 1



III. DATA

The signals used in the "Line Echo Cancellation" project can be represented as follows:
Impulse Response Sequence: The impulse response sequence of the echo path is denoted as h[n]. It is a real-world sequence loaded from the file path.mat.
Composite Source Signal: The composite source signal, representing speech-like behavior, is denoted as CSS[n]. It is a synthetic signal loaded from the file css.mat. CSS[n] contains segments of pause, periodic excitation, and white noise properties.
These signals play a crucial role in various aspects of the project. The impulse response sequence, h[n], is used for analyzing the impulse and frequency responses of the echo

path. The composite source signal, CSS[n], is used for generating echo signals and evaluating the performance of the adaptive line echo canceller.
By using these symbolic representations, the project can perform calculations, analyses, and simulations involving these signals to develop and evaluate the line echo cancellation system effectively.

IV. Math

A. Equations

the unknown system is time invariant, which indicate that the coefficients of its impulse response are constants and finite such that the desired response is given by

$$d[n] = \sum_{k=0}^{M-1} h[k]x[n-k]$$

The output of an adaptive FIR filter with the same number of coefficients, M, is given by

$$y[n] = \sum_{k=0}^{M-1} w[k]x\,[n-k]$$

For these two systems to be equal, the difference e[n]=d[n]-y[n] must be equal to zeros. It is the method of adaptive filtering that will enable us to learn the system coefficients and produce an error e[n], approximately equal zero.

V. Evaluation Criteria

The "Line Echo Cancellation" project will be evaluated based on several criteria. These include the successful implementation of the adaptive line echo canceller system, its performance in mitigating echo interference and improving voice quality, the accuracy and thoroughness of the analysis conducted, clear visualization and presentation of results, effective teamwork and collaboration, a well-structured and informative project report, proficient presentation and demonstration skills, and the level of creativity and innovation shown in addressing the problem. The evaluation aims to assess the team's technical skills, analytical abilities, communication, and overall project performance.

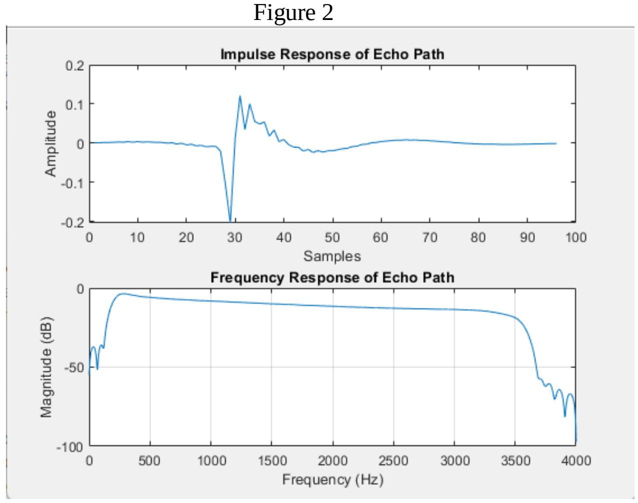
VI. Approach

The project followed a systematic approach to address the problem of line echo cancellation. It began by understanding the nature of line echo and analyzing the impulse response sequence of the echo path. Synthetic signals were generated to emulate speech-like behavior for evaluation. The adaptive line echo canceller was implemented using the normalized least mean squares (NLMS) algorithm. The performance of the system was evaluated using objective metrics such as echo return loss (ERL), mean square error (MSE), and signal-to-noise ratio (SNR). Additionally, alternative adaptive algorithms were explored and compared to the NLMS approach. This approach ensured a comprehensive understanding of the problem, effective echo cancellation, and the identification of optimal techniques.

VII. Results and Analysis

a) Plot the impulse and frequency responses of the echo path.

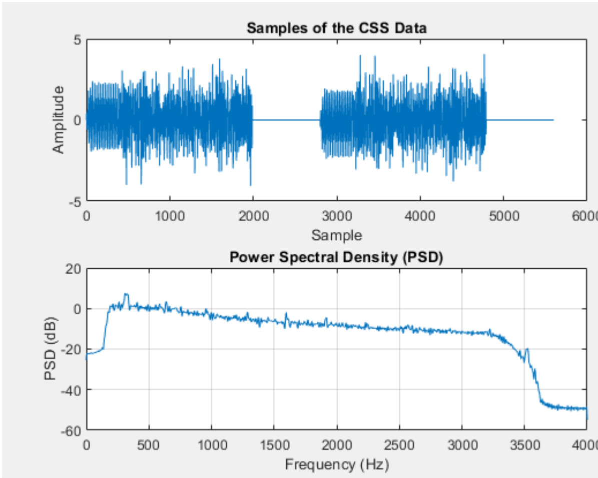
Figure 2



The figure provides a comprehensive visualization of the echo path, enabling examination of its behavior in both the time and frequency domains. This facilitates an understanding of how the echo path influences the input signal.

b) Plot the samples of the CSS data, as well as their spectrum (Power Spectrum Density PSD).

Figure 3



The figure presents a complete representation of the CSS data, including both its temporal and frequency features. The behavior and spectral makeup of the composite source signal are easier to comprehend via to this depiction.

c) Plot the resulting echo signal.

Figure 4

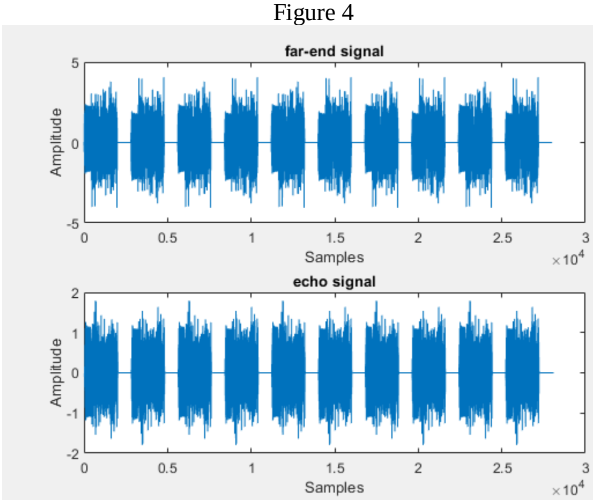


Figure 5

```
Input Signal Power: 0.00 dB
Output Signal Power: -6.35 dB
Echo-Return-Loss (ERL): -6.35 dB
Input Signal Power: 2.9e-15 dB
Output Signal Power: -6.3 dB
Echo-Return-Loss (ERL): -6.3 dB
```

The output values obtained from the code represent the

input power: **10*log10(1/N * sum(abs(css_signal).^2))** ,

output power: 10*log10(1/N_output * sum(abs(echo).^2)),

and **echo-return-loss (ERL) of the echo path: output_power - input_power**. The input power is the energy or strength of the input signal, which is calculated from the concatenated and repeated CSS signal. The output power is the energy or strength of the echo signal obtained after convolving the input signal with the impulse response. The ERL represents the attenuation introduced by the echo path, indicating how much weaker the echo signal is compared to the input signal. Positive ERL values indicate attenuation, while negative values indicate amplification. These values provide insights into the power characteristics and the effectiveness of echo cancellation or suppression in the echo path.

d) Plot the far-end signal, the echo, and the error signal provided by the adaptive filter. Plot also the echo path and its estimate by the adaptive filter at the end of the simulation.

Figure 6

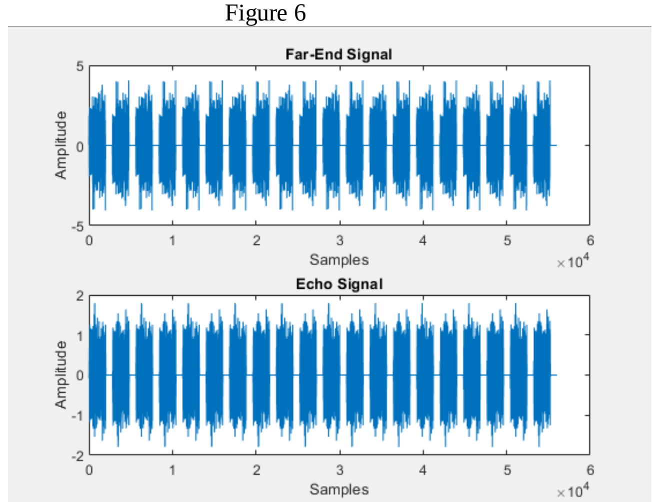


Figure 7

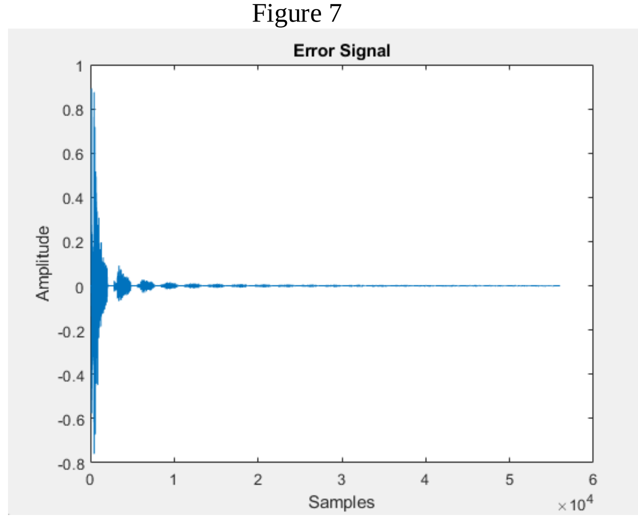


Figure 8

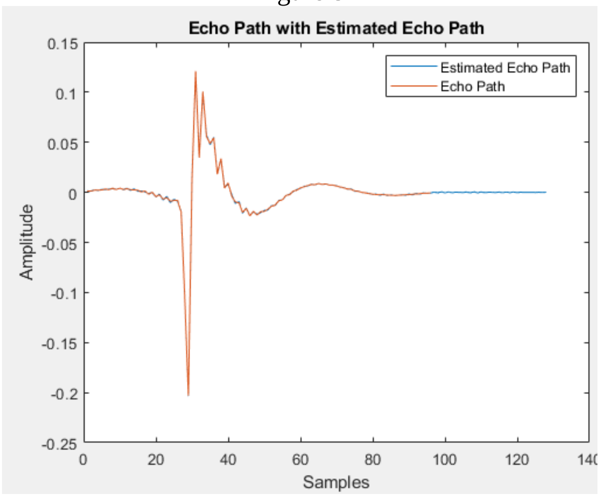


Figure 6:

- Subplot 1: Displays the far-end signal, which represents the input signal to the echo path.
- Subplot 2: Shows the echo signal, which represents the output signal of the echo path.

Figure 7:

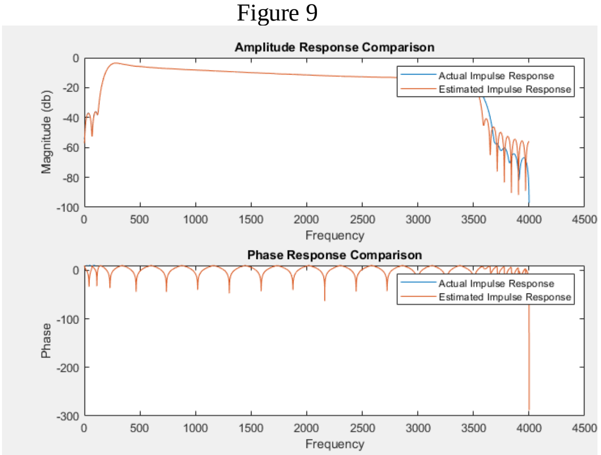
- Plots the error signal, which represents the difference between the estimated echo and the actual echo signal.

Figure 8:

- Plots the estimated echo path (adaptive filter coefficients) overlaid with the actual echo path.

e) Plot the amplitude and phase response for the estimated FIR channel at the end of the iterations. Compare it with the given FIR system (Path).

Figure 9



f) Propose a different appropriate Adaptive algorithm and compare it to the ∈-NLMS

Figure 10

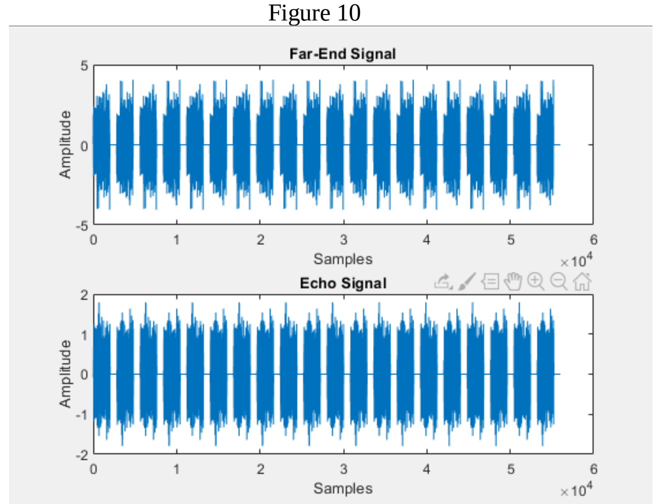
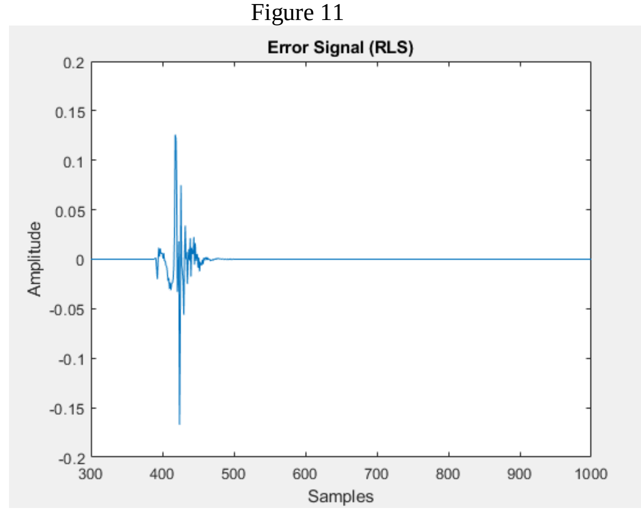


Figure 11



These figures provide a visual representation of the performance of the RLS algorithm in estimating and canceling out the echo. By examining the far-end signal, echo signal, and error signal, the effectiveness of the algorithm in removing the echo component can be evaluated.

I. CONCLUSION

the project aimed to understand and analyze the properties of an echo path. The project involved loading the impulse response sequence of the echo path and visualizing it through plots of the impulse and frequency responses.

The impulse response plot showed how the echo path behaves over time, indicating the delays and attenuations introduced by the path. It helped us understand how the echo evolves and fades over time.

The frequency response plot provided insights into how the echo path affects different frequencies. It allowed us to see which frequencies are amplified or attenuated by the path, giving us an understanding of its frequency-dependent characteristics.

By concatenating blocks of a composite source signal and feeding them into the echo path, we simulated the echo effect and analyzed the resulting echo signal. We estimated the input and output powers of the signal and calculated the echo-return-loss (ERL) as the difference between them, indicating the level of attenuation caused by the echo path.

Overall, the project provided valuable knowledge about the behavior of the echo path, helping us understand how it affects signals in terms of time and frequency. This understanding is useful in various applications where echo management is important, such as audio processing and telecommunications.

II. References

[1]https://www.diva-portal.org/smash/get/diva2:280596/fulltext01

[2] https://en.wikipedia.org/wiki/Least_mean_squares_filter

[3]https://www.keil.com/pack/doc/CMSIS/DSP/html/group__LMS.html

III. Appendix

The Full Code:

```
clear
load('path.mat');
impulse_response = path;

figure;

subplot(2, 1, 1);
plot(impulse_response);
xlabel('Samples');
ylabel('Amplitude');
title('Impulse Response of Echo Path');

subplot(2, 1, 2);
Fs = 8000;
freq = 0:Fs/2;
freq_response = freqz(impulse_response, 1, freq, Fs);
magnitude_db = 20*log10(abs(freq_response)); %
Convert magnitude to dB
plot(freq, magnitude_db);
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
title('Frequency Response of Echo Path');
grid on;


%-------------------------------------------------------------
load('css.mat');
css_signal = css;

figure;
subplot(2, 1, 1);
plot(css_signal);
xlabel('Sample');
ylabel('Amplitude');
title('Samples of the CSS Data');

subplot(2, 1, 2);
psd = pwelch(css_signal);
freq_range = linspace(0, Fs/2, length(psd));
plot(freq_range, 10*log10(psd));
xlabel('Frequency (Hz)');
ylabel('PSD (dB)');
title('Power Spectral Density (PSD)');
grid on;


%-------------------------------------------------------------


css_signal = css_signal(:);
far_end = repmat(css_signal,5, 1);

impulse_response = impulse_response(:);

echo = conv(far_end, impulse_response);

figure;
subplot(2, 1, 1);
plot(far_end);
xlabel('Samples');
ylabel('Amplitude');
title('far-end signal');

subplot(2, 1, 2);
plot(echo);
xlabel('Samples');
ylabel('Amplitude');
title('echo signal');
% Calculate the power in dB for the input and output
signals


% Calculate the power in dB for the input signal
N = length(css_signal); % Length of the input signal
```

```
input_power = 0;
for i = 1:N
   input_power = input_power + abs(css_signal(i))^2;
end
input_power = 10*log10(1/N * input_power);

% Calculate the power in dB for the output signal
N_output = length(echo); % Length of the output
signal
output_power = 0;
for i = 1:N_output
   output_power = output_power + abs(echo(i))^2;
end
output_power = 10*log10(1/N_output *
output_power);
ERL =  output_power-input_power ;
fprintf('Input Signal Power: %.2f dB\n',
input_power);
fprintf('Output Signal Power: %.2f dB\n',
output_power);
fprintf('Echo-Return-Loss (ERL): %.2f dB\n', ERL);
fprintf('Input Signal Power: %.2g dB\n',
input_power);
fprintf('Output Signal Power: %.2g dB\n',
output_power);
fprintf('Echo-Return-Loss (ERL): %.2g dB\n', ERL);


%-------------------------------------------------------------

far_end = repmat(css_signal, 10, 1);
echo = filter(impulse_response, 1, far_end);

mu = 0.25;
epsilon = 1e-6;
m = 128;
w = zeros(m, 1);
estimated_echo = zeros(size(far_end));
e = zeros(size(far_end));

for i = 1:length(far_end)
   if i >= m
      x = far_end(i:-1:i-m+1);
      y = w' * x;
      error = echo(i) - y;
      w = w + (mu / (norm(x)^2 + epsilon)) * error *
x;
      estimated_echo(i) = y;
      e(i) = error;
   end
end

figure;
subplot(2,1,1);
plot(far_end);
xlabel('Samples');
ylabel('Amplitude');
title('Far-End Signal');

subplot(2,1,2);
plot(echo);
xlabel('Samples');
ylabel('Amplitude');
title('Echo Signal');

figure;
plot(e);
xlabel('Samples');
ylabel('Amplitude');
title('Error Signal');

figure;
plot(w);
hold on;
plot(impulse_response);
xlabel('Samples');
ylabel('Amplitude');
legend('Estimated Echo Path','Echo Path');
title('Echo Path with Estimated Echo Path');
%---------------------------------------------------------------
adapted_response = freqz(w, 1, freq, Fs);
db = 20*log10(abs(adapted_response)); % Convert
magnitude to dB
figure;
subplot(2,1,1);
plot(magnitude_db);
hold on;
plot(db);
xlabel('Frequency');
ylabel('Magnitude (db)');
legend('Actual Impulse Response', 'Estimated
Impulse Response');
title('Amplitude Response Comparison');


 db1= 20*log10(abs(angle(impulse_response))); %
Convert magnitude to dB
 db2= 20*log10(abs(angle(adapted_response))); %
Convert magnitude to dB
subplot(2,1,2);
plot(db1);
hold on;
plot(db2);
xlabel('Frequency');
ylabel('Phase');
legend('Actual Impulse Response', 'Estimated
Impulse Response');
title('Phase Response Comparison');


%-----------------------------------------------------------

la = 0.99;
delta = 1e-3;
forgetting = 1 / la;

w_rls = zeros(m, 1);
j_rls = zeros(size(far_end));
e_rls = zeros(size(far_end));
```

```
P = eye(m) / delta;

for i = 1:length(far_end)
    if i >= m
    x = far_end(i:-1:i-m+1);
    y = w_rls' * x;
    e_rls(i) = echo(i) - y;

    k = (P * x) / (la + x' * P * x);
    P = (1 / la) * (P - k * x' * P);
    w_rls = w_rls + k * e_rls(i);
    j_rls(i) = y;

    end
end

figure;
subplot(2,1,1);
plot(far_end);
xlabel('Samples');
ylabel('Amplitude');
title('Far-End Signal');

subplot(2,1,2);
plot(echo);
xlabel('Samples');
ylabel('Amplitude');
title('Echo Signal');

figure;
plot(e_rls);
xlabel('Samples');
ylabel('Amplitude');
title('Error Signal (RLS)');
ylim([-0.2, 0.2]);
xlim([300, 1000]);
```