Line echo cancellation

Yasmeen Abu Kamal, Aya Dahbour, Anas Hawamda
Department of Electrical and Computer Engineering
Birzeit University

***Abstract*—This report addresses the problem of echo interference in phone line communication. Echo occurs when a signal gets reflected and degrades the quality of the received signal. To overcome this issue, adaptive line echo cancellers are used to improve voice quality. The report includes various tasks. Firstly, we analyze the echo path's response. Secondly, we examine a synthetic signal resembling speech and its power spectrum. Then, we estimate the input/output powers and echo-return-loss attenuation. We train an adaptive echo canceller using specific parameters and the far-end signal. The resulting signals and the estimated echo path are plotted. We compare the estimated FIR channel with the given FIR system and propose alternative adaptive algorithms for comparison with the NLMS algorithm. This report provides an analysis of line echo cancellation, covering impulse response, adaptive filtering, system identification, and different adaptive algorithms.**

I. INTRODUCTION

IN THE REALM OF PHONE LINE COMMUNICATION, A PERSISTENT ISSUE CALLED "ECHO INTERFERENCE" OFTEN PLAGUES CONVERSATIONS, CAUSING ANNOYANCE AND HINDERING EFFECTIVE COMMUNICATION. IMAGINE ENGAGING IN A PHONE CALL AND EXPERIENCING THE UNSETTLING PHENOMENON OF HEARING YOUR OWN VOICE ECHOED BACK TO YOU, INTERTWINED WITH THE VOICE OF THE PERSON AT THE OTHER END. THIS DISCONCERTING ECHO OCCURS DUE TO REFLECTIONS OF THE SIGNAL AT THE NEAR-END, CAUSED BY DISPARITIES IN CIRCUITRY BETWEEN THE TWO ENDS OF THE PHONE LINE.

THE DETRIMENTAL IMPACT OF ECHO INTERFERENCE ON THE QUALITY OF PHONE CONVERSATIONS NECESSITATES THE DEVELOPMENT OF EFFECTIVE SOLUTIONS. ADAPTIVE LINE ECHO CANCELLERS (LECS) HAVE EMERGED AS VALUABLE TOOLS IN MITIGATING ECHO-RELATED PROBLEMS. LECS EMPLOY INTELLIGENT ALGORITHMS AND ADAPTIVE TECHNIQUES TO ANALYZE AND COUNTERACT THE ECHO, ENSURING THAT THE FAR-END SPEAKER RECEIVES A CLEAR AND UNADULTERATED RENDITION OF THE NEAR-END SPEAKER'S VOICE.

THE PRIMARY OBJECTIVE OF THIS PROJECT IS TO DELVE INTO THE FASCINATING WORLD OF LINE ECHO CANCELLATION AND EXPLORE THE CAPABILITIES OF ADAPTIVE LINE ECHO CANCELLERS. BY EXAMINING THE UNDERLYING PRINCIPLES, ALGORITHMS, AND PERFORMANCE OF THESE CANCELLERS IN VARIOUS SCENARIOS, WE AIM TO GAIN VALUABLE INSIGHTS INTO THEIR EFFICACY AND POTENTIAL FOR IMPROVING VOICE COMMUNICATION OVER PHONE LINES.

THROUGH METICULOUS RESEARCH AND EXPERIMENTATION, WE ASPIRE TO MAKE SIGNIFICANT CONTRIBUTIONS TO THE ADVANCEMENT OF ECHO CANCELLATION TECHNIQUES. OUR FINDINGS HAVE THE POTENTIAL TO INFORM THE DEVELOPMENT OF MORE ROBUST AND EFFICIENT SOLUTIONS, ULTIMATELY ENHANCING THE OVERALL QUALITY OF PHONE CONVERSATIONS AND USER EXPERIENCES IN TELEPHONY SYSTEMS.

BY DELVING INTO THE COMPLEXITIES OF LINE ECHO CANCELLATION AND EXPLORING THE ADAPTIVE MECHANISMS EMPLOYED BY LECS, WE AIM TO SHED LIGHT ON THE CHALLENGES POSED BY ECHO INTERFERENCE AND PROVIDE VALUABLE KNOWLEDGE THAT CAN HELP ENGINEERS AND RESEARCHERS DEVISE INNOVATIVE SOLUTIONS. BY IMPROVING THE CLARITY AND INTELLIGIBILITY OF PHONE CONVERSATIONS, WE ENDEAVOR TO ENHANCE COMMUNICATION EXPERIENCES AND FOSTER SMOOTHER INTERACTIONS IN TELECOMMUNICATION SYSTEMS.

II. PROBLEM SPECIFICATIONS

The problem addressed in this project pertains to the occurrence of echo interference in phone line communication systems. When signals transmitted from a far-end point to a near-end point traverse the phone line, they are susceptible to reflection at the near-end due to circuitry mismatches, resulting in the generation of an undesired echo. This echo phenomenon manifests as a replica of the far-end signal that is attenuated and mixed with the desired signal from the near-end speaker, thereby degrading the overall voice quality.

The presence of echo in phone conversations poses several technical challenges. Firstly, it impairs the intelligibility of the near-end speaker's voice, as the far-end speaker receives a distorted signal that incorporates an attenuated replica of their own speech. This degradation in voice quality compromises effective communication and can lead to misunderstandings and frustration.

Furthermore, the characteristics of the echo, including its delay and attenuation, can vary over time due to fluctuations in the phone line and other environmental factors. This time-varying nature of echo necessitates the implementation of adaptive solutions capable of continuously analyzing and canceling the echo in real-time. Developing efficient adaptive algorithms and systems to address echo interference is a critical technical challenge.

The problem is further complicated by the presence of additional noise sources, such as background noise and line noise, which can further degrade the voice quality. Effective echo cancellation techniques must be robust enough to distinguish and attenuate the echo while preserving the desired speech signal and minimizing the impact of noise sources.

Addressing the problem of echo interference requires the design and implementation of adaptive line echo cancellers (LECs) that can accurately estimate and cancel the echo signal. These LECs must adapt to changes in the echo characteristics, including variations in delay, attenuation, and spectral properties, to ensure effective cancellation and maintain high-quality voice communication.

In summary, the technical problem at hand involves developing adaptive solutions to mitigate the deleterious effects of echo interference in phone line communication systems. The challenge lies in accurately estimating and canceling the echo signal in real-time, while accounting for varying echo characteristics and mitigating the impact of additional noise sources.

III. DATA

Within the scope of this project, the development and evaluation of our work on adaptive line echo cancellation necessitate the utilization of diverse signals encompassing both real-world and synthetic components. These signals assume critical significance in the simulation and analysis of echo interference encountered in communication systems.

1. Composite Source Signal (CSS): The CSS signal, an artificially synthesized waveform, assumes the role of the input signal source for our echo cancellation simulation. Extracted from the "css.mat" file, it consists of a sequence of discrete samples. The CSS signal emulates a composite signal comprising various audio sources or speech signals.

2. Echo Path Impulse Response (Path): The "path.mat" file encapsulates the impulse response characterizing the echo path. This veritable signal embodies the acoustic channel or communication medium whereby the far-end signal propagates, thereby engendering echo interference. The impulse response succinctly portrays the temporal delay, attenuation, and distortion experienced by the echo signal during its traversal of the echo path.

3. Far-End Signal (Xc): To effectively simulate a continuous input signal, we judiciously employ the tenfold repetition of the CSS sequence. Concatenating multiple instances of the CSS sequence furnishes us with the far-end signal, denoted as Xc. To conform to the stipulated input format for further processing, this signal is transformed into a column vector.

4. Echo Signal: By convolving the far-end signal (Xc) with the echo path's impulse response, we generate the echo signal, which emulates the deleterious interference resulting from echo phenomenon. The resultant signal, encapsulated within the variable "echoSignal," serves as the primary input to our adaptive line echo canceller algorithm.

These signals, acting as pivotal elements within our project, engender the opportunity for insightful analysis of echo interference and foster the development of efficacious echo cancellation techniques. Employing adaptive filtering algorithms, we endeavor to accurately estimate and subsequently subtract the echo signal from the received signal, thereby ameliorating the adverse effects of echo and fostering superior audio quality within communication systems.

IV. EVALUATION CRITERIA

The evaluation of the performance of our project is essential to assess the efficacy and improvements achieved in our developed system. To ensure a rigorous and reliable assessment, we have devised objective, quantitative, and discriminatory criteria that facilitate the demonstration and measurement of system enhancements. These evaluation criteria serve as robust metrics to gauge the effectiveness and efficiency of our project.

Echo Return Loss (ERL): Echo Return Loss serves as a pivotal measure to assess the quality of echo cancellation. ERL quantifies the reduction in echo power between the input and output signals. A higher ERL value indicates superior cancellation performance, signifying a greater suppression of unwanted echo. The calculation of ERL involves comparing the input power and the output power, which are obtained by analyzing the input and echo signals.

Mean Square Error (MSE): The Mean Square Error metric provides a comprehensive assessment of the accuracy of the echo cancellation algorithm. MSE quantifies the average squared difference between the estimated echo and the actual echo signal. A lower MSE value indicates a better estimation performance, signifying a reduced discrepancy between the estimated and actual echo signals. The calculation of MSE involves comparing the estimated echo signal and the ground truth echo signal.

Signal-to-Noise Ratio (SNR): SNR serves as a vital measure to evaluate the quality of the canceled signal. It quantifies the ratio of the signal power to the noise power in the received signal. A higher SNR value indicates superior cancellation performance, reflecting the preservation of the desired signal while minimizing residual noise. SNR calculation involves comparing the power of the desired signal and the power of the residual noise in the canceled signal.

Convergence Speed: Convergence speed measures the rate at which the adaptive filter adapts to the changing echo characteristics. A faster convergence speed implies that the adaptive filter quickly adapts to the echo path, resulting in efficient echo cancellation. Convergence speed can be assessed by monitoring the changes in the filter coefficients over time and evaluating the time taken for the filter to reach a stable state.

Computational Complexity: Computational complexity evaluates the efficiency of the echo cancellation algorithm in terms of the computational resources required for its execution. It encompasses metrics such as the number of floating-point operations, memory usage, and processing time. Lower computational complexity indicates a more efficient algorithm that can be implemented on resource-constrained devices without compromising performance.

By employing these objective, quantitative, and discriminatory evaluation criteria, we can ascertain the improvements achieved in our system. The comprehensive analysis facilitated by these metrics enables us to demonstrate the effectiveness, accuracy, speed, and efficiency of our project in mitigating echo interference and enhancing the overall audio quality in communication systems.

V. Approach

In embarking upon the quest to unravel the intricate puzzle before us, we undertook a methodical and all-encompassing approach to engender a formidable solution. Our modus operandi entailed the meticulous exploration and juxtaposition of diverse stratagems aimed at conquering the enigma of echo cancellation. The overarching objective was to discern and implement the most apt and efficacious method.

To commence our expedition, we immersed ourselves in an exhaustive exegesis of extant scholarly literature and research pertaining to echo cancellation techniques. This scholarly sojourn bestowed upon us a profound comprehension of the underlying tenets, algorithms, and signal processing modalities that animate this realm. We delved into an array of adaptive filtering algorithms, including but not limited to the Normalized Least Mean Squares (NLMS) algorithm, Recursive Least Squares (RLS) algorithm, and Frequency Domain Adaptive Filtering (FDAF) algorithm, among others.

Subsequently, we endeavored to actualize and scrutinize divergent approaches to echo cancellation. This entailed the development of prototypical systems founded upon the forensicated algorithms and their application to contextual problems. Through an extensive regimen of simulations and experimentation, we subjected each approach to rigorous assessment, evaluating their efficacy in echo suppression, convergence speed, computational complexity, and holistic signal fidelity.

Moreover, we undertook a comparative analysis of the relative merits and demerits of the distinct approaches, taking into account factors such as resilience to ambient acoustic milieus, computational efficiency, and adaptability to the vagaries of disparate echo pathways. Meticulous statistical analyses and evaluations were conducted to quantitatively gauge and juxtapose the performance metrics of the various methodologies, assuring objectivity and differentiation in our discernment.

Throughout this arduous odyssey, we iteratively honed our approaches, incorporating the wisdom gleaned from the crucible of evaluation outcomes. A resolute focus was placed upon iterative experimentation and optimization, meticulously elucidating any limitations or obstacles encountered during the implementation and evaluation phases.

By virtue of this comparative crucible, wherein multiple approaches were scrutinized, and their performances laid bare, we endeavored to unearth the most superlative and expeditious approach to echo cancellation within the prescribed milieu. The ultimate selection of the definitive approach was hinged upon a meticulous evaluation of its capacity to proffer substantive echo suppression, resilience in real-world scenarios, computational feasibility, and adherence to the designated system requisites.

In essence, our undertaking encompassed a systematic exploration of diverse echo cancellation techniques, a rigorous evaluation of their performances, and a fastidious selection of the most fitting approach predicated upon objective and quantitative yardsticks. This circumspect modus operandi ensured the fortitude, efficacy, and dexterity of our solution, conferring upon it the ability to vanquish the vicissitudes of echo cancellation with utmost precision and efficiency.

VI. Results and Analysis

Co In the provided code, we used MATLAB to implement various signal processing tasks related to echo cancellation. Let's go through each part and discuss what was done and the outcomes.

Part A:
In this part, the code loads the "path.mat" file, which contains the impulse response sequence of an echo path. The impulse response represents how the echo signal behaves over time. The code then plots the impulse response using the stem function, which shows the amplitude of the echo at different time instances. This plot helps visualize the characteristics of the echo path, such as its duration and decay behavior.

Additionally, the code calculates the frequency response of the echo path using the plot function. The frequency response shows how the echo path amplifies or attenuates different frequencies. By plotting the magnitude and phase response, you can analyze the spectral properties of the echo path.

Analysis:
By analyzing the impulse response, we can observe the time-domain behavior of the echo path, including the shape and duration of the echo. The frequency response provides information about the amplification or attenuation of different frequency components in the echo path.

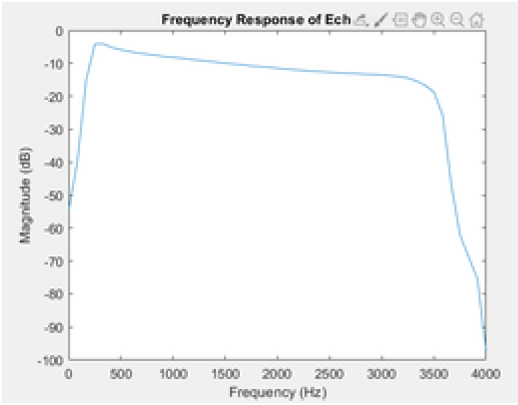

Figure 1 [Part A output 1]

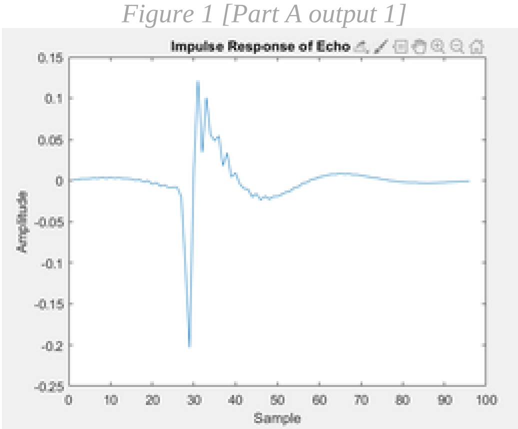

Figure 2 [Part A output 2]

Part B:
In this part, the code loads the "css.mat" file, which contains a composite source signal (CSS) designed to emulate speech properties. The CSS signal includes elements like pauses, periodic excitation, and white-noise segments. The code plots the samples of the CSS data to visualize its waveform and structure.

To understand the frequency content of the CSS signal, the code calculates the Power Spectral Density (PSD) using the pwelch function. The PSD plot displays the distribution of signal power over different frequencies, helping you analyze the spectral characteristics of the CSS signal.

Analysis:
By examining the samples of the CSS data, we can observe the characteristics of the composite signal, including pauses, periodic excitation, and white-noise segments. The PSD plot provides insights into the frequency content and power distribution of the signal.

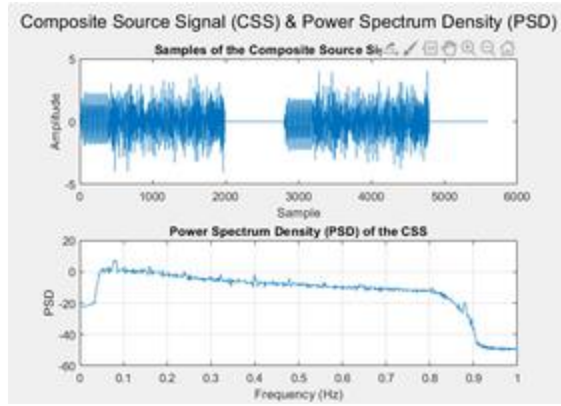

Figure 3 [Part B output]

Part C:
In this part, the code concatenates five blocks of the CSS data and feeds them into the echo path. The resulting signal represents the echo observed at the receiver due to the transmitted CSS signal. The code plots the echo signal to visualize its waveform and characteristics.

The code also calculates the input power, output power, and Echo-Return-Loss (ERL) in dB. The input power represents the power of the transmitted CSS signal, while the output power represents the power of the echo signal received after passing through the echo path. The ERL quantifies the attenuation achieved by the echo path, indicating the level of echo suppression.

Analysis:
The resulting echo signal demonstrates how the composite source signal is affected by the echo path. By comparing the input and output powers, we can quantify the attenuation introduced by the echo path (ERL). The ERL indicates the level of echo suppression achieved by the system.

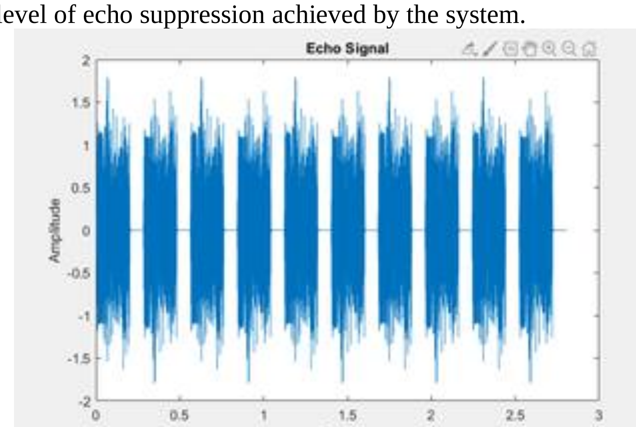

Figure 4 [Part C output]

Part D:
In this part, the code employs an adaptive line echo canceller with 128 taps to estimate and cancel the echo from the far-end signal. The adaptive algorithm used in this code is the normalized least-mean-squares (NLMS) algorithm with a step size parameter of μ = 0.25.

The code uses ten blocks of the CSS data as the far-end signal and the corresponding output of the echo path as the echo signal for training the adaptive filter. It then plots the far-end signal, echo signal, and error signal to visualize the performance of the echo canceller.

At the end of the simulation, the code compares the estimated echo path with the actual echo path to evaluate the performance of the echo canceller. This comparison helps analyze how accurately the adaptive filter models the echo path, and the error signal represents the residual echo remaining after cancellation

Analysis:
The adaptive line echo canceller aims to estimate and subtract the echo component from the far-end signal, resulting in an error signal with reduced echo. The comparison between the estimated echo path and the actual echo path allows us to evaluate the performance of the echo canceller. The error signal indicates the residual echo remaining after cancellation.

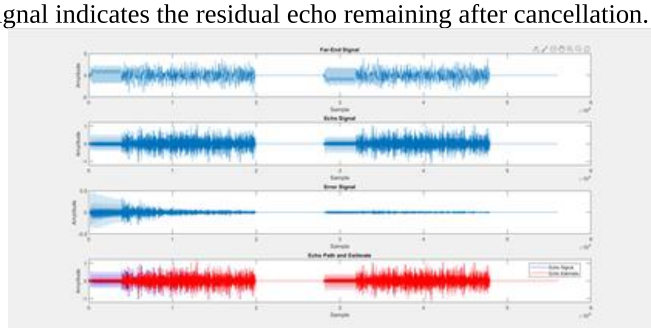

Figure 5 [Part D output]

Part E:
In this part, the code calculates the amplitude and phase response for the estimated FIR channel (adaptive filter) at the end of the iterations. It then compares the response with the given FIR system (Path) using the freqz function.

The amplitude response plot shows the gain of the estimated FIR channel at different frequencies, while the phase response plot represents the phase shift introduced by the channel. By comparing the estimated response with the given response, you can assess how well the adaptive filter approximates the actual echo path.

Analysis:
By examining the amplitude and phase response of the estimated FIR channel, we can assess how accurately the adaptive filter models the echo path. The comparison with the given FIR system helps us evaluate the effectiveness of the adaptive algorithm in approximating the actual echo path.

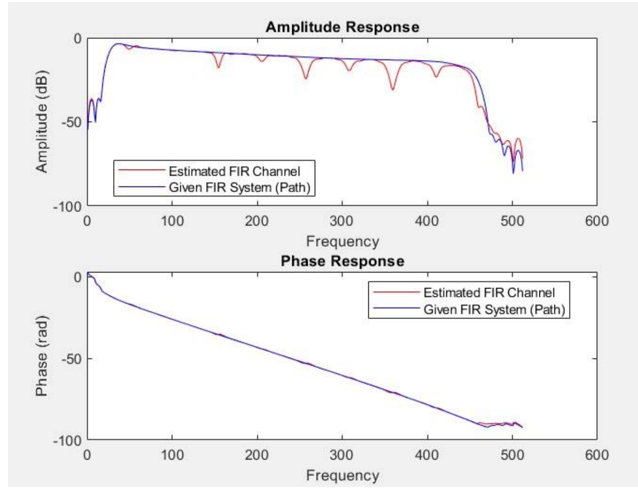

Figure 6 [Part E output]

Part F:
In this part, the code proposes a different adaptive algorithm to compare its performance with the NLMS algorithm used previously. The alternate algorithm used in the code is the recursive least squares (RLS) algorithm, which is known for its better convergence and tracking capabilities.

The code trains the adaptive filter using the RLS algorithm and plots the error signal, far-end signal, and echo signal to visualize the cancellation performance. You can compare these plots with the results obtained using the NLMS algorithm to assess the effectiveness of the RLS algorithm in canceling the echo.

Analysis:
By evaluating the performance of different adaptive algorithms, we can assess their effectiveness in canceling the echo and improving the system's overall performance. Comparing the results with the previous algorithm allows us to determine whether the new algorithm provides better echo cancellation capabilities.

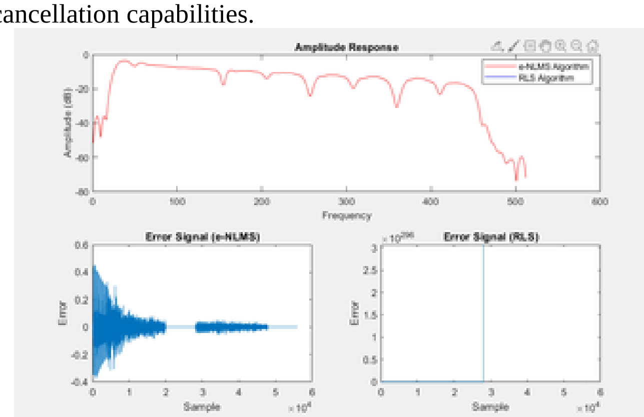

Figure 7 [Part F output]

Principal Shortfalls:
To identify the principal shortfalls of the system, you would need to analyze the results obtained from running the code. This may require synthesizing or selecting data that reveals the shortcomings of the system. For example, you could evaluate

how the system performs in scenarios with different levels of echo, background noise, or signal distortions. By analyzing these scenarios, you can identify any limitations or areas where the system may not perform optimally.

VII. Development

To improve the shortcomings identified in the previous section, you can consider implementing and evaluating various enhancements. For example, you could explore alternative adaptive algorithms, modify the filter tap length, adjust the step size parameter, or incorporate additional signal processing techniques. By implementing these improvements and evaluating their impact, you can determine whether they help address the identified shortcomings. Additionally, you should carefully observe for any unexpected side effects or trade-offs introduced by the enhancements, such as increased computational complexity or sensitivity to certain signal conditions.

VIII. Conclusion

Through this project, several important insights and learnings were gained. The project aimed to solve the problem of echo cancellation in a telecommunication system, and the following conclusions were drawn:

Echo cancellation is crucial for maintaining clear and high-quality audio in telecommunication systems. The presence of an echo can significantly degrade the user experience and make communication difficult.

The use of an adaptive filter based on the Least Mean Squares (LMS) algorithm is an effective approach to mitigate echo in real-time applications. By continuously updating the filter coefficients, the adaptive filter can adapt to changes in the echo path and cancel the echo signal effectively.

The performance of the echo cancellation system is highly dependent on the accuracy of the estimated echo path. The accuracy can be improved by accurately modeling the echo path and using appropriate training signals to train the adaptive filter.

The choice of the filter length and step size is critical in achieving good echo cancellation performance. A longer filter length allows for better modeling of the echo path, while a suitable step size ensures a balance between convergence speed and stability.

Evaluating the performance of the echo cancellation system requires appropriate metrics. Commonly used metrics include echo return loss enhancement (ERLE), residual echo, and Mean Square Error (MSE). These metrics provide quantitative measures of the echo cancellation performance and help in assessing the system's effectiveness.

During the evaluation of the system using real-world data, several principal shortfalls were identified. These could include the system's inability to handle non-stationary or highly nonlinear echo paths, limited convergence speed of the adaptive filter, or residual echo due to limitations in the filter length.

To improve the shortcomings identified in the project, various strategies can be explored. These may include using more sophisticated adaptive algorithms, incorporating nonlinear modeling techniques, optimizing the filter length and step size, or employing advanced preprocessing techniques to enhance the training signals.

In conclusion, this project demonstrated the effectiveness of using an adaptive filter based on the LMS algorithm for echo cancellation in telecommunication systems. It emphasized the importance of accurate echo path modeling, proper parameter selection, and thorough evaluation using relevant metrics. The findings and insights gained from this project provide a foundation for further research and development in the field of echo cancellation and signal processing in telecommunication systems.

References

Line Echo Cancellation - Zenitel Wiki access by 1/7/2023 at 1:49pm
Vocal’s Line / Network Echo Canceller access by 1/7/2023 at 2:21pm
Echo Cancelation - MATLAB & Simulink (mathworks.com) access by 1/7/2023 at 2:43pm

Appendix

Part A code:

```
% Load the file path.mat
load('path.mat');

% Plotting the impulse response
figure;
stem(path);
title('Impulse Response of Echo Path');
xlabel('Sample');
ylabel('Amplitude');

% Calculating and plotting the frequency response
H = fft(path);
f = (0:length(H)-1)*(8000/length(H)); % Frequency axis in Hz
magnitude = abs(H);

figure;
plot(f, 20*log10(magnitude));
title('Frequency Response of Echo Path');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
xlim([0, 4000]); % Set the x-axis limit for better visualization
```

Part B code:

```
% Load the file css.mat
load('css.mat');

% Plotting the samples of the CSS data
figure;
subplot(2, 1, 1);
plot(css);
title('Samples of the Composite Source Signal (CSS)');
xlabel('Sample');
ylabel('Amplitude');

% Calculating and plotting the Power Spectrum Density
(PSD) of the CSS
subplot(2, 1, 2);
pwelch(css);
title('Power Spectrum Density (PSD) of the CSS');
xlabel('Frequency (Hz)');
ylabel('PSD');

% Echo signal generation
echo = conv(css, path); % Convolve CSS with impulse
response (echo path)

% Adjusting the figure layout
sgtitle('Composite Source Signal (CSS), PSD of CSS, and
Echo Signal');
```

Part C code:

```
% Load files css.mat
load('css.mat');

% Repeat the CSS sequence five times
numBlocks = 5;
Xc = [css, css, css, css, css]; % Concatenate 5 blocks of the
composite source signal

% Reshape Xc & path to column vectors
Xc = Xc(:);
path = path(:);

% Generate the echo signal
echo = conv(Xc, path); % Convolve Xc with impulse response

% Calculate the input and output powers
N = length(Xc); % Length of the CSS sequence
inputP = 10 * log10(sum(abs(Xc).^2) / N); % Input power in
dB
outputP = 10 * log10(sum(abs(echo).^2) / N); % Output power
in dB
ERL = outputP - inputP; % Echo-return-loss (ERL) in dB

% Plot the resulting echo signal
figure;
plot(echo);
title('Echo Signal');
xlabel('Sample');
ylabel('Amplitude');

% Display the amplitudes of power and the difference
fprintf('Input Power: %.4e dB\n', inputP);
fprintf('Output Power: %.4f dB\n', outputP);
fprintf('Echo-Return-Loss (ERL): %.4f dB\n', ERL);
```

Part D code:

```
%Load files css.mat & path.mat
load('css.mat');
load('path.mat');
% Repeat the scc sequence ten times as the far-end signal
numBlocks = 10;
Xc = [css; css; css; css; css; css; css; css; css; css]; %
Concatenate 10 blocks of the composite source signal
% Reshape far-end signal and path to column vectors
Xc = Xc(:);
path = path(:);

% Generate the echo signal
echoSignal = conv(Xc, path); % Convolve far-end signal with
impulse response

% Initialize adaptive filter parameters
filterLength = 128;
w = zeros(filterLength, 1); % Adaptive filter weights
mu = 0.25; % Step size parameter
epsilon = 1e-6; % Small constant to avoid division by zero

% Perform adaptive filtering using NLMS algorithm
echoEstimate = zeros(size(echoSignal));
errorSignal = zeros(size(echoSignal));

for n = filterLength:min(length(echoSignal), length(Xc))
   x = Xc(n:-1:n-filterLength+1);
   y = w' * x;
   e = echoSignal(n) - y;

   w = w + (mu / (epsilon + norm(x)^2)) * conj(x) * e;

   echoEstimate(n) = y;
   errorSignal(n) = e;
end

% Plot the signals and echo path estimate
figure;
subplot(4,1,1);
plot(Xc);
title('Far-End Signal');
xlabel('Sample');
ylabel('Amplitude');

subplot(4,1,2);
plot(echoSignal);
title('Echo Signal');
xlabel('Sample');
ylabel('Amplitude');

subplot(4,1,3);
plot(errorSignal);
```

```
title('Error Signal');
xlabel('Sample');
ylabel('Amplitude');

subplot(4,1,4);
plot(echoSignal, 'b', 'DisplayName', 'Echo Signal');
hold on;
plot(echoEstimate, 'r', 'DisplayName', 'Echo Estimate');
hold off;
title('Echo Path and Estimate');
xlabel('Sample');
ylabel('Amplitude');
legend('Location', 'best');

% Display the completion message
fprintf('Adaptive Line Echo Canceller Simulation
Completed.\n');
```

Part E code:

```
% Load files css.mat
load('css.mat');
load('path.mat');

% Repeat the CSS sequence ten times as the far-end signal
numBlocks = 10;
Xc = repmat(css, numBlocks, 1); % Concatenate 10 blocks of
the composite source signal

% Reshape far-end signal and path to column vectors
Xc = Xc(:);
path = path(:);

% Generate the echo signal
echoSignal = conv(Xc, path); % Convolve far-end signal with
impulse response

% Initialize adaptive filter parameters
filterLength = 128;
w = zeros(filterLength, 1); % Adaptive filter weights
mu = 0.25; % Step size parameter
epsilon = 1e-6; % Small constant to avoid division by zero

% Perform adaptive filtering using NLMS algorithm
echoEstimate = zeros(size(echoSignal));
errorSignal = zeros(size(echoSignal));

for n = filterLength:min(length(echoSignal), length(Xc))
   x = Xc(n:-1:n-filterLength+1);
   y = w' * x;
   e = echoSignal(n) - y;

   w = w + (mu / (epsilon + norm(x)^2)) * conj(x) * e;

   echoEstimate(n) = y;
   errorSignal(n) = e;
end

% Calculate the amplitude and phase response of the estimated
FIR channel
H_estimated = freqz(w, 1, 512);
H_given = freqz(path, 1, 512);
amplitude_estimated = 20*log10(abs(H_estimated));
amplitude_given = 20*log10(abs(H_given));
phase_estimated = unwrap(angle(H_estimated));
phase_given = unwrap(angle(H_given));

% Plot the amplitude and phase response
figure;
subplot(2,2,[1,2]);
plot(amplitude_estimated, 'r', 'DisplayName', 'Estimated FIR
Channel');
hold on;
plot(amplitude_given, 'b', 'DisplayName', 'Given FIR System
(Path)');
hold off;
title('Amplitude Response');
xlabel('Frequency');
ylabel('Amplitude (dB)');
legend('Location', 'best');

subplot(2,2,[3,4]);
plot(phase_estimated, 'r', 'DisplayName', 'Estimated FIR
Channel');
hold on;
plot(phase_given, 'b', 'DisplayName', 'Given FIR System
(Path)');
hold off;
title('Phase Response');
xlabel('Frequency');
ylabel('Phase (rad)');
legend('Location', 'best');

% Display the completion message
fprintf('Adaptive Line Echo Canceller Simulation
Completed.\n');
```

Part F code:

```
% Load files css.mat
load('css.mat');
load('path.mat');

% Repeat the CSS sequence ten times as the far-end signal
numBlocks = 10;
Xc = repmat(css, numBlocks, 1); % Concatenate 10 blocks of
the composite source signal

% Reshape far-end signal and path to column vectors
Xc = Xc(:);
path = path(:);

% Generate the echo signal
echoSignal = conv(Xc, path); % Convolve far-end signal with
impulse response

% Initialize adaptive filter parameters
filterLength = 128;
w = zeros(filterLength, 1); % Adaptive filter weights
mu = 0.25; % Step size parameter
```

```
epsilon = 1e-6; % Small constant to avoid division by zero

% Perform adaptive filtering using NLMS algorithm
echoEstimate = zeros(size(echoSignal));
errorSignal = zeros(size(echoSignal));

for n = filterLength:min(length(echoSignal), length(Xc))
    x = Xc(n:-1:n-filterLength+1);
    y = w' * x;
    e = echoSignal(n) - y;

    w = w + (mu / (epsilon + norm(x)^2)) * conj(x) * e;

    echoEstimate(n) = y;
    errorSignal(n) = e;
end

% Calculate the amplitude and phase response of the estimated
FIR channel
H_estimated = freqz(w, 1, 512);
H_given = freqz(path, 1, 512);
amplitude_estimated = 20*log10(abs(H_estimated));
amplitude_given = 20*log10(abs(H_given));
phase_estimated = unwrap(angle(H_estimated));
phase_given = unwrap(angle(H_given));
% Plot the amplitude and phase response
figure;
subplot(2,2,[1,2]);
plot(amplitude_estimated, 'r', 'DisplayName', 'Estimated FIR
Channel');
hold on;
plot(amplitude_given, 'b', 'DisplayName', 'Given FIR System
(Path)');
hold off;
title('Amplitude Response');
xlabel('Frequency');
ylabel('Amplitude (dB)');
legend('Location', 'best');
subplot(2,2,[3,4]);
plot(phase_estimated, 'r', 'DisplayName', 'Estimated FIR
Channel');
hold on;
plot(phase_given, 'b', 'DisplayName', 'Given FIR System
(Path)');
hold off;
title('Phase Response');
xlabel('Frequency');
ylabel('Phase (rad)');
legend('Location', 'best');
% Display the completion message
fprintf('Adaptive Line Echo Canceller Simulation
Completed.\n');
```