Line Echo Cancellation

Nidal Zabade
Electrical and Computer Engineering
Birzeit University
nidal1zabade@gmail.com

Mahmoud Shihab
Electrical and Computer Engineering
Birzeit University
sandi.pc.man@gmail.com

Mohammad Abu-Shelbaia
Electrical and Computer Engineering
Birzeit University
mabushelbaia@gmail.com

***Abstract*—This paper explores the problem of echo interference in phone line communications and the efficiency of adaptive LECs in lowering echo interference. A mismatch in circuitry causes an echo generation which causes the recipient to receive a replica of itself in addition to the received signal. An adaptive line echo canceller (LEC)is used to overcome this problem. Adaptive LECs ensure clear phone conversations by employing the far-end signal as input and its reflected version as a reference.**

I. Introduction

Clear and high-quality audio during phone conversations is dependent on effective signal transmission in communication networks. Signal reflections and echoes are frequently caused by circuitry inconsistencies such as hybrid connections, however. The desired audio is interfered with by these echoes, which are brought on by the reflected signals returning to the far-end point, lowering the overall speech quality. [1] Adaptive line echo cancellers (LEC) are frequently used to improve speech quality at both ends of the transmission to solve this problem. These cancellers employ adaptive algorithms to continually modify their filter coefficients in response to the features of the echo route in order to reduce or completely eradicate the echo. We concentrate on the issue of line echo cancellation in this research. Our goal is to create and evaluate an adaptive line echo canceller for communication systems to reduce echo impacts. This aim is accomplished through a number of important phases and tasks that are part of the project. To improve speech quality in communication systems, the overall goal of this research is to design and study an adaptive line echo canceller. We want to identify efficient ways to reduce the effects of line echoes in phone calls by comprehending the features of the echo path, examining the composite source signal, training the adaptive canceller, and comparing various methods. [1]

II. Problem Specification

The existence of line echoes in communication networks, notably phone lines, is the issue this research attempts to solve. Due to circuitry mismatches, such as hybrid connections, a signal that is traveling from the far end to the near end will reflect at the near end. The targeted audio signal is subsequently interfered with when this reflected signal returns to the far-end point as an echo. The received signal quality is greatly reduced by the line echoes, making it challenging for the listener at the far end to comprehend the speaker at the near end. This issue is more common during phone interactions and can lead to a breakdown in communication, decreased productivity, and user unhappiness. This project's goal is to create an adaptive line echo canceller (LEC) that successfully reduces line echoes and enhances speech quality in communication networks. The canceller tries to eliminate unwanted echoes and improve the quality of the received audio by continually adjusting its filter coefficients based on the features of the echo route by using adaptive algorithms. In order to accomplish this goal, it is essential to solve a number of major issues:

1) Echo route Analysis: For an echo canceller to work well, it is essential to understand the features of the echo route. To understand the echo path's characteristics, such as delay, attenuation, and frequency-dependent behavior, it is necessary to evaluate the impulse response sequence and frequency response of the echo path.
2) Composite Source Signal Analysis: The project's composite source signal (CSS) simulates speech features and includes pauses, periodic excitation, and parts that resemble white noise. Understanding the CSS's frequency content and reproducing real-world voice sounds may be accomplished by analyzing the PSD, or power spectrum density.
3) Echo Signal Estimation: By concatenating CSS data blocks and sending them into the echo route, the resulting echo signal may be estimated. The signal levels prior to and during echo cancellation may be measured quantitatively by estimating the input and output powers in dB.
4) Selection and Training of Adaptive Algorithms: The success of the echo canceller depends on the selection of a suitable adaptive algorithm. This project makes use of the -NLMS algorithm with certain settings. The far-end signal must be used as the input data for the adaptive filter and the echo signal must be used as the reference data. It is necessary to assess the adaptive filter's convergence and performance. [2]
5) Comparative Analysis: To evaluate the performance of the created adaptive line echo canceller, the -NLMS algorithm is compared to an alternate adaptive method. Their effectiveness, rate of convergence, and canceling capacities are compared. [2]

The project seeks to resolve the issue of line echoes in communication systems by addressing these difficulties. The created adaptive line echo canceller has the ability to greatly enhance speech quality by cutting down on interference brought on by echoes and improving all aspects of communication.

III. Data

A 5600-sample composite source signal (CSS) is the source of the data for this project. This artificial signal is ideal for studying and testing echo cancellation techniques since it was created expressly to mimic the characteristics of speech. Different segments that replicate various features of speech are included into the CSS signal. It comprises pauses, which stand for breaks in speech or moments of stillness. The signal also includes periodic excitation portions that mimic the voiced sounds made during speech. These recurring excitation intervals follow a predictable pattern and add to the rhythmic quality of speech. The CSS signal also contains portions that have white-noise characteristics, simulating the turbulent or unvoiced speech sounds. The CSS data's 5600 samples are collected at an industry-standard 8 kHz sampling rate for telecommunications. This sample rate enables efficient analysis and processing while ensuring that the signal faithfully replicates the original analog voice signal. Multiple components of the project depend on the CSS data. It acts as the signal input used to drive the echo route and create the echo signal. Concatenating CSS data blocks allows for the estimation of the echo signal, which makes it possible to assess the effectiveness of echo cancellation. In order to comprehend the frequency content and properties of the synthetic speech signal, the PSD of the CSS data is also examined.

Overall, the composite source signal's 5600 samples give the information required to model and examine the impacts of line echoes in communication systems. The research seeks to produce an efficient adaptive line echo canceller that may improve sound quality in telecommunication applications using this synthetic signal that mimics the characteristics of speech.

IV. Evaluation Criteria

1) Graphical Representation: Using graphics to show the error signal, voice signal, and echo signal is a crucial assessment criteria. It is simpler to study these signals' properties, spot any lingering echoes, and gauge how well the echo cancellation methods work when the signals are plotted across time.
2) Echo Return Loss (ERL): ERL is a crucial statistic for assessing how well echo canceling devices operate. It measures how much the canceller was able to reduce echo power. The designed adaptive line echo canceller's effectiveness in reducing echoes and enhancing voice quality may be evaluated by calculating the ERL.
3) Comparative Analysis of Adaptive Algorithms: Comparing various adaptive algorithms, such as the Normalized Least Mean Square (NLMS) algorithm and the Normalized Least Mean Fourth (NLMF) algorithm, is another criteria for evaluation. We can assess how well each algorithm cancels the echoes by analyzing the error signals it generates. To determine which algorithm is best for a certain application, it is also possible to evaluate variables such algorithm convergence speed and stability.
4) Software Development: An important assessment criterion is creating software implementations for the various operations on the signals. The program should include features for loading and processing input signals, using adaptive line echo canceller methods, and producing the required graphical analysis representations. The simplicity and effectiveness, and correctness of the software implementation will have a role in how the project is judged overall.

We may evaluate the performance, effectiveness, and efficiency of the created adaptive line echo canceller by taking into account these assessment criteria. The project's effectiveness in reducing line echoes and enhancing speech quality in communication systems will be shown through a combination of graphical representations, ERL calculations, a comparative study of adaptive algorithms, and software implementation.

V. Approach

1) Part One
   - Load the impulse response of the channel from the file "path.mat".
   - Calculate the frequency response of the channel using sp.signal.freqz.
   - Plot the impulse response and frequency response of the channel using plt.plot.
   - Display the plots using plt.show.
2) Part Two
   - Load the speech signal from the file "css.mat".
   - Plot the speech signal using plt.plot.
   - Calculate and plot the Power Spectral Density (PSD) of the speech signal using plt.psd.
   - Display the plots using plt.show.
3) Part Three
   - Load the impulse response of the channel from the file "path.mat".
   - Load the speech signal from the file "css.mat".
   - Concatenate the speech signal 5 times to get the Far-End signal.
   - Convolve the speech signal with the impulse response of the channel to get the echo signal.
   - Plot the Far-End signal, echo signal, and the Power Density Spectrum (PSD) of the echo signal.
   - Calculate the power of the Far-End signal and echo signal.
   - Calculate the Echo Return Loss (ERL) using the power values.
   - Display the plots and ERL value.
4) Part Four
   - Load the impulse response of the channel from the file "path.txt".
   - Load the speech signal from the file "css.txt".

- Concatenate the speech signal 10 times to get the Far-End signal.
- Convolve the speech with the impulse response to get the echo signal.
- Define the filter length and the step size for the adaptive filter.
- Initialize arrays for the error signal and estimated echo signal.
- Create an adaptive filter using pa.filters.FilterNLMS.
- Adapt the filter coefficients using the Far-End signal and echo signal.
- Calculate the error signal by subtracting the estimated echo signal from the echo signal.
- Plot the Far-End signal, echo signal, error signal, and estimated echo signal.
- Display the plots.

5) Part Five

- Load the impulse response of the channel from the file ”path.txt”.
- Load the speech signal from the file ”css.txt”.
- Concatenate the speech signal 10 times to get the Far-End signal.
- Convolve the speech with the impulse response to get the echo signal.
- Define the filter length and the step size for the adaptive filter.
- Initialize arrays for the error signal and estimated echo signal.
- Create an adaptive filter using pa.filters.FilterNLMS.
- Adapt the filter coefficients using the Far-End signal and echo signal.
- Calculate the error signal by subtracting the estimated echo signal from the echo signal.
- Plot the amplitude response and phase response for the estimated FIR channel.
- Display the plots.

6) Part Six

- Load the impulse response of the channel from the file ”path.txt”.
- Load the speech signal from the file ”css.txt”.
- Concatenate the speech signal 10 times to get the Far-End signal.
- Convolve the speech with the impulse response to get the echo signal.
- Define the filter length and the step size for the adaptive filters (NLMS and NLMF).
- Initialize arrays for the error signals and estimated echo signals.
- Create adaptive filters using pa.filters.FilterNLMS and pa.filters.FilterNLMF.
- Adapt the filter coefficients using the Far-End signal and echo.

VI. Results and Analysis

A. Part One

In this part, we plot the impulse response and frequency response of the channel where the path.mat is the impulse response.

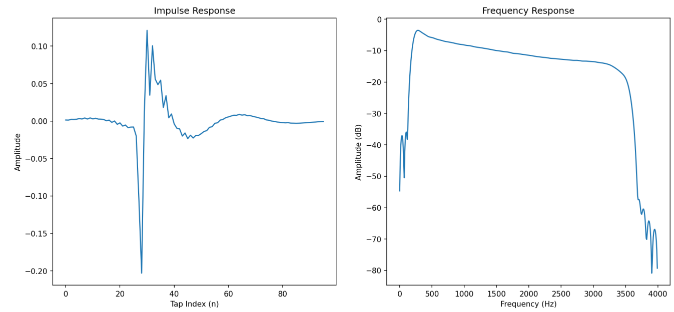

Fig. 1. Path signal and its frequency response

B. Part Two

In this part, we calculate and plot the Power Spectral Density (PSD) of the speech signal where path.mat is the speech signal.

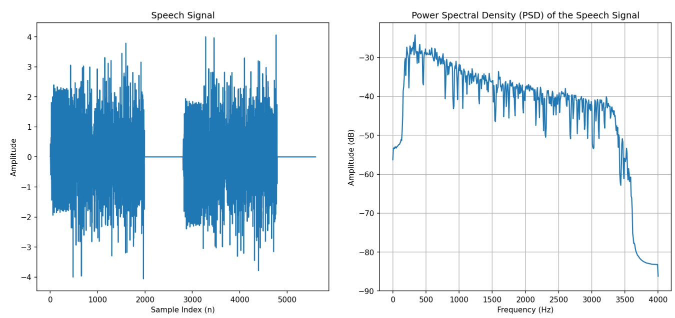

Fig. 2. Speech signal and its power density spectrum

C. Part Three

In this part, we convolve the concatenated signal five times with the impulse response then plot it and calculate the power of the Far-End signal and echo signal, and calculate the Echo Return Loss (ERL) using the power values.

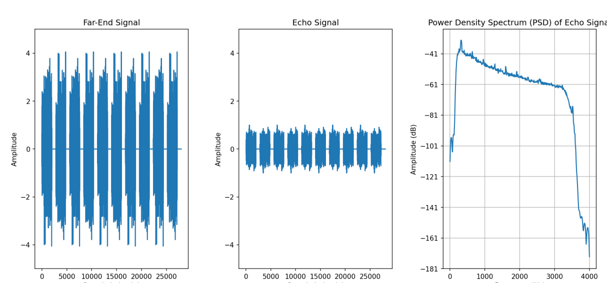

Fig. 3. Far-end signal, echo signal, and the power density spectrum for the echo

The power of the Far-End signal: -9.643274665532882e-15 The power of the echo signal: -11.4178994674944 The ERL: 11.417899467494392

D. Part Four

In this part, we do the same as part three and we create an adaptive filter using the NLMS algorithm then adapt the filter coefficients using the Far-End signal and echo signal then Calculate the error signal by subtracting the estimated echo signal from the echo signal then plot. As we can see the

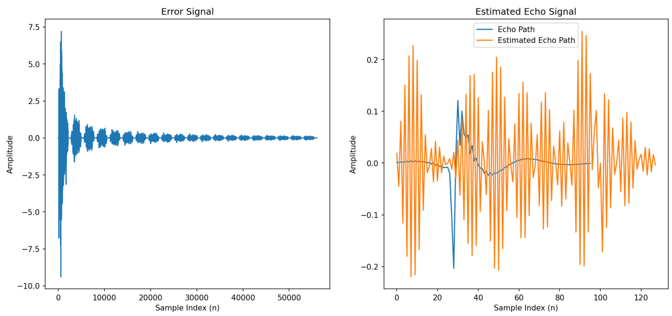

Fig. 4. Error signal using NLMS and the echo signal vs estimated echo signal

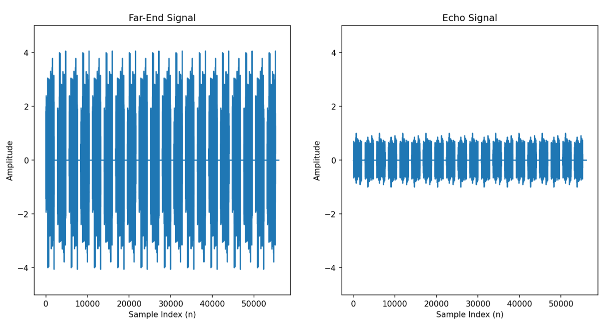

Fig. 5. Echo signal vs estimated echo signal

error is decreasing very fast which means that this technique is good.

E. Part Five

This part is the same as part 4 but instead, we plot the amplitude and phase response for the path and the estimated path for the FIR channel.

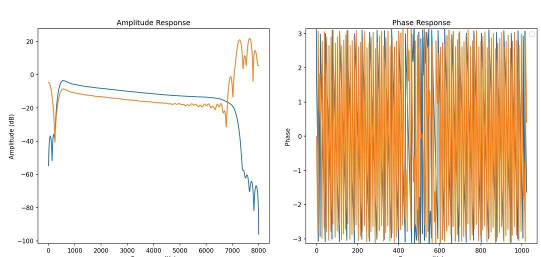

Fig. 6. Far-end and echo signals

F. Part Six

In this part we will compare two algorithms for the adaptive filter the first is the one we use NLMS and the second one is NLMF and check what algorithm is better than the other. As

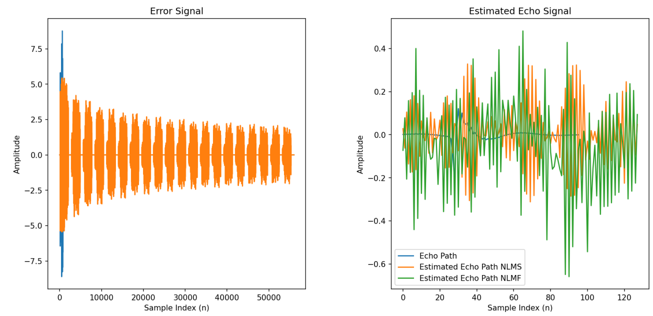

Fig. 7. Amplitude and the phase response for the echo signal vs estimated echo signal

we can see that the first algorithm "NLMS" is better than the second one "NLMF" even though we put mio in the second one much smaller than the first one and the error signal in the first one decreased faster than the other one, also the estimated Echo from the first one is closer.

VII. Conclusion

In order to increase speech quality, we tackled the issue of line echoes in communication systems in this project and created an adaptive line echo canceller. We've produced big results and learned a lot through in-depth study, simulations, and assessments. First, we looked at the impulse response pattern and frequency response of the echo route to examine its properties. The design and execution of the adaptive canceller were influenced by this analyses' improved understanding of the echo route and its behavior. Second, to recreate real-world speech signals, we used a composite source signal (CSS) that mimics speech features. We learned more about the frequency composition and characteristics of the synthetic speech signal by examining the CSS and its power spectrum density (PSD). We approximated the resultant echo signal by feeding blocks of the signal into the echo route using the CSS data. This gave us the opportunity to assess the echo canceller's effectiveness and measure the input and output powers in dB. The signal levels prior to and during echo cancellation were quantified in these evaluations. Using the far-end signal as the input and the echo signal as the reference, we used the -NLMS method and trained the adaptive filter. We investigated the convergence and functionality of the adaptive filter by putting the method into practice with certain settings. The error signals, convergence rates, and cancellation abilities of the -NLMS method and another adaptive algorithm were also examined.

References

[1] Matić, V. M., & Abadžić, S. N. (2007). Acoustic and line echo cancellation using adaptive filters. November 20-22, 2007.

[2] M. Matoušek, "Normalized Least-mean-fourth (NLMF) - Padasip 1.2.1 documentation," Matousc89, Jul. 2, 2023. https://matousc89.github.io/padasip/sources/filters/nlmf.html (accessed Jul. 2, 2023).