Project Report: Echo Generator for Signal Processing

Prepared By:
Amira Sherif 202300076
Mohammed Maher 202301837

Supervisor:
Dr. Moustafa Elshafei

Zewail University of Science and Technology
Faculty of Communications and Information Engineering

Project Report: Echo Generator for Signal Processing

Amira Sherif (ID: 202300076)
Mohammed Maher (ID: 202301837)

Introduction

The Echo Generator is a fundamental project in signal processing designed to introduce the concept of delayed signal processing. The primary goal is to manipulate an input audio signal by adding a delayed version of itself to create an echo effect. This report outlines the project's methodology, implementation, and results, along with test cases to demonstrate the system's functionality and versatility.

Objective

The main objective is to implement an echo generator using MATLAB, where the output signal is a combination of the input signal and its delayed version. The delay and the amplitude of the echo are adjustable parameters, allowing for experimentation with different effects.

Methodology

The echo generator operates based on the following equation:

$$Y[n] = x[n] - \alpha \cdot Y[n - N_d]$$

Where:

- $Y[n]$: Output signal.
- $x[n]$: Input signal.
- α: Echo amplitude factor (0.4 to 0.6).
- N_d: Delay in samples, calculated as:

$$N_d = F_s \cdot \text{(Delay in seconds)}$$

Steps:

1. Read the input audio signal.
2. Convert stereo signals to mono if necessary.
3. Configure parameters for delay (in seconds) and echo amplitude.
4. Generate the output signal by applying the echo equation iteratively.
5. Normalize the output signal to prevent clipping.
6. Export the processed signal as an audio file.
7. Validate the results by listening to both the original and processed signals.

Implementation

MATLAB Code

The code was implemented as follows:

```
% Read input audio file
[x, Fs] = audioread('input_file.mp3');
if size(x, 2) > 1
    x = mean(x, 2);  % Convert to mono
end

% Parameters for echo effect
alpha = 0.5;            % Echo amplitude
delay_seconds = 0.1;  % Delay time in seconds
Nd = round(Fs * delay_seconds);  % Convert delay to
   samples

% Initialize output array
y = zeros(length(x) + Nd, 1);

% Copy input signal to output
y(1:length(x)) = x;

% Generate echo effect
for n = (Nd + 1):length(y)
    y(n) = (n <= length(x)) * x(n) - alpha * y(n - Nd);
end
```

```
% Normalize output
y = y / max(abs(y));

% Write output to file
audiowrite('output.mp3', y, Fs);

% Play original and echo sounds
sound(x, Fs);
pause(length(x)/Fs + 1);
sound(y, Fs);
```

Test Cases and Results

Test Case 1

- **Parameters:**
  - Input Signal: Speech recording (10 seconds).
  - α: 0.4.
  - Delay: 0.1 seconds.
- **Expected Result:** A subtle echo effect, with the echo slightly quieter than the original signal.
- **Outcome:** The processed signal exhibited a clear echo, validating the implementation.

Test Case 2

- **Parameters:**
  - Input Signal: Music clip (15 seconds).
  - α: 0.6.
  - Delay: 0.5 seconds.
- **Expected Result:** A pronounced echo with a moderate delay, emphasizing the echo effect.
- **Outcome:** The processed signal demonstrated the expected effect, highlighting the flexibility of the code.

Test Case 3

- **Parameters:**
  - Input Signal: White noise (10 seconds).
  - α: 0.5.
  - Delay: 2 seconds.
- **Expected Result:** A long echo with a consistent delay, creating a reverberation-like effect.
- **Outcome:** The system successfully added the delayed signal without artifacts, confirming robustness.

Discussion

The results confirm that the echo generator is effective in applying delayed signal processing. The implementation supports a range of inputs and parameter configurations. Key observations include:

1. Lower α values produce subtle effects, while higher values amplify the echo.
2. Increasing the delay enhances the perceived separation between the original signal and the echo.
3. Normalization ensures audio quality remains unaffected by signal amplitude.

Conclusion

This project successfully implemented an echo generator using MATLAB. The system proved capable of processing various audio signals and producing the desired effects by adjusting the echo parameters. This exercise reinforced key signal processing concepts, such as time-domain manipulation, iterative algorithms, and normalization. Relevant references are included below to support the methodologies described in this report.

References

1. Oppenheim, A. V., & Schafer, R. W. (2009). *Discrete-Time Signal Processing*. Pearson.

2. MATLAB Documentation: audioread and audiowrite.

3. Lyons, R. G. (2010). *Understanding Digital Signal Processing*. Pearson.