Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

MoE, UGC & AICTE Approved

NAAC A++ Accredited

Division of Electronics and Communication Engineering
2023-2024 (EVEN SEM)

III IA EVALUATION REPORT

for

DIGITAL SIGNAL PROCESSING-PROJECT BASED COURSE

Title of the project

ACTIVE NOISE CONTROL

A report submitted by

Name of the Student	***Thimothy Prasanna S***
Register Number	***URK22EC4012***
Subject Name	***Digital Signal Processing***
Subject Code	***18EC2015***
Date of Report submission	***05/04/2024***

Project Rubrics for Evaluation

First Review: Project title selection - PPT should have four slides (Title page, Introduction, Circuit/Block Diagram, and Description of Project).

Second Review: PPT should have three slides (Description of Concept, implementation, outputs, results and discussion)

Rubrics for project (III IA - 40 Marks):
Content - 4 marks (based on Project)
Clarity - 3 marks (based on viva during presentation)
Feasibility - 3 marks (based on project)
Presentation - 10 marks
Project Report - 10 marks
On-time submission - 5 marks (before the due date)
Online submission-GCR - 5 marks

Total marks:______/ 40 Marks

Signature of Faculty with date:

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In today's bustling world, noise pollution is a prevalent issue that can disrupt our daily lives, affecting productivity, comfort, and overall well-being. Active Noise Control (ANC) technology offers a promising solution by actively reducing unwanted noise in various environments. To make ANC accessible and user-friendly, we introduce the Active Noise Control GUI.

What is ANC?

Active Noise Control (ANC) is a sophisticated technology that uses sound waves to reduce or eliminate unwanted noise in a given space. By analyzing ambient noise and generating anti-noise signals, ANC systems can effectively cancel out undesirable sounds, creating a quieter and more peaceful environment for users.

The Need for ANC GUI

While ANC technology holds great potential, its implementation can be complex and challenging for non-experts. The ANC GUI simplifies the process by providing an intuitive interface for users to apply noise cancellation to audio files effortlessly. Whether you're a professional sound engineer or a casual user, the ANC GUI offers a convenient tool to enhance your audio experience.

CHAPTER 2

DESCRIPTION OF THE PROJECT

This project presents a MATLAB-based Graphical User Interface (GUI) designed to facilitate Active Noise Control (ANC). ANC is a technique employed to reduce unwanted sound by generating an anti-noise signal that interferes destructively with the noise, thereby diminishing its impact.

The GUI interface allows users to select two audio files: the original audio, containing both desired signal and noise, and the noise audio, which solely contains unwanted noise. Upon selection of both files, the application executes noise cancellation by subtracting the noise audio from the original audio sample by sample. The resultant signal, representing the noise-cancelled audio, is displayed in a distinct waveform plot.

Key features of the GUI include three axes dedicated to visualizing the original audio waveform, noise audio waveform, and the output (noise-cancelled) audio waveform. Additionally, users can interact with the GUI through buttons, enabling actions such as selecting audio files, applying noise cancellation, and listening to the noise-cancelled audio.

By providing a user-friendly interface for real-time exploration of ANC techniques, this project serves as an invaluable tool for comprehending the fundamental principles of noise cancellation and its practical application in scenarios such as audio recording, communication systems, and noise-sensitive environments.

CHAPTER 3

CONCEPT INVOLVED

GUI Creation: The code creates a GUI using MATLAB's figure and uicontrol functions. The GUI consists of buttons for selecting the original audio file, noise audio file, and applying noise cancellation. Additionally, there are three axes for displaying the original audio waveform, noise audio waveform, and the output (noise-cancelled) audio waveform.

Audio Input: The application allows the user to select two audio files: the original audio containing both the desired signal and noise, and the noise audio containing only the noise to be canceled. This is achieved using the uigetfile function to open file selection dialogs.

Noise Cancellation: When the "Apply Noise Cancellation" button is clicked, the code checks if both the original and noise audio files have been selected. If so, it performs noise cancellation by subtracting the noise audio from the original audio.

This operation is done sample by sample, effectively creating an anti-noise signal. The result is stored in the outputAudio variable.

Audio Visualization: The original audio waveform, noise audio waveform, and the output audio waveform are plotted in separate axes using the plot function. This allows the user to visualize the effect of noise cancellation.

Audio Playback: After noise cancellation, the output audio is played using the sound function, allowing the user to listen to the noise-cancelled audio.

Sampling Rate: The code assumes a sampling rate of 44100 Hz for the audio files. This sampling rate is commonly used for CD-quality audio.

In summary, the code demonstrates a basic implementation of ANC through a MATLAB GUI application. It provides a user-friendly interface for selecting audio files, performing noise cancellation, visualizing the audio waveforms, and listening to the noise-cancelled audio. While this implementation is simplistic and may not achieve perfect noise cancellation in all scenarios, it serves as a starting point for understanding and experimenting with ANC techniques.

CHAPTER 4

TOOLS

Tools Utilized in the Active Noise Control (ANC) GUI Implementation

This MATLAB-based GUI for Active Noise Control (ANC) leverages several tools and functionalities to provide users with a seamless experience in selecting audio files, applying noise cancellation, and visualizing the results. Below is a comprehensive description of the tools involved:

MATLAB GUI Components:

The GUI is built using MATLAB's GUI components, such as figures, axes, and pushbuttons. These components are arranged on the interface to create a user-friendly environment for ANC operations.

Global Variables:

Global variables originalAudio, noiseAudio, and outputAudio are declared to store audio data throughout the application. This allows data to persist across different functions and enables efficient communication between GUI elements.

Audio Processing Functions:

The application utilizes MATLAB's built-in audio processing functions, such as audioread for reading audio files and sound for playing back audio. These functions facilitate audio input/output operations and enable real-time playback of noise-cancelled audio.

Plotting Functions:

MATLAB's plotting functions, including plot and axis, are employed to visualize audio waveforms in the GUI. Waveforms of the original audio, noise audio, and noise-cancelled audio are displayed on separate axes, providing users with a clear understanding of the ANC process.

File Selection Dialogs:

File selection dialogs are implemented using MATLAB's uigetfile function. These dialogs allow users to browse and select audio files from their file system, enabling them to input both the original audio and noise audio for ANC processing.

Mathematical Operations:

Basic mathematical operations, such as subtraction and array indexing, are performed to apply noise cancellation to the original audio. The noise audio is subtracted from the original audio sample by sample to generate the noise-cancelled audio.

User Interaction:

User interaction is facilitated through pushbuttons with callback functions. When users click on buttons such as "Select Original Audio," "Select Noise Audio," and "Apply Noise Cancellation," corresponding actions are triggered, enabling users to interact with the GUI seamlessly.

Audio Sampling Rate:

A sampling rate of 44100 Hz is assumed for the audio files, which is commonly used for CD-quality audio. This sampling rate is applied consistently throughout the application to ensure accurate audio processing and playback.

CHAPTER 5

CODE IMPLEMENTATION

```
% Create a function for the GUI
function ANC_GUI()
    % Create a figure
    fig = figure('Position', [200, 200, 1000, 600], 'MenuBar', 'none', 'Name', 'Active
Noise Control');

    % Declare global variables to store audio data
    global originalAudio;
    global noiseAudio;
    global outputAudio;
    % Create axes for original audio waveform
    ax1 = axes('Units', 'pixels', 'Position', [50, 400, 300, 150]);
    xlabel('Time (s)');
    ylabel('Amplitude');
    title('Original Audio');

    % Create axes for noise audio waveform
    ax2 = axes('Units', 'pixels', 'Position', [400, 400, 300, 150]);
    xlabel('Time (s)');
    ylabel('Amplitude');
    title('Noise Audio');

    % Create axes for output audio waveform
    ax3 = axes('Units', 'pixels', 'Position', [750, 400, 300, 150]);
    xlabel('Time (s)');
    ylabel('Amplitude');
    title('Noise Cancellation Audio');

    % Create UI elements
    uicontrol('Style', 'pushbutton', 'String', 'Select Original Audio', 'Position', [150,
300, 150, 30], 'Callback', @selectOriginalAudio);
    uicontrol('Style', 'pushbutton', 'String', 'Select Noise Audio', 'Position', [500, 300,
150, 30], 'Callback', @selectNoiseAudio);
    uicontrol('Style', 'pushbutton', 'String', 'Apply Noise Cancellation', 'Position',
[850, 300, 150, 30], 'Callback', @applyNoiseCancellation);

    % Function to select original audio file
    function selectOriginalAudio(~, ~)
        [file, path] = uigetfile('*.wav', 'Select Original Audio');
        if file
            originalAudio = audioread(fullfile(path, file));
            disp('Original audio file selected');
            % Plot original audio waveform
            plot(ax1, (1:length(originalAudio)) / 44100, originalAudio);
            axis(ax1, 'tight');
        end
    end

    % Function to select noise audio file
    function selectNoiseAudio(~, ~)
        [file, path] = uigetfile('*.wav', 'Select Noise Audio');
        if file
            noiseAudio = audioread(fullfile(path, file));
            disp('Noise audio file selected');
            % Plot noise audio waveform
            plot(ax2, (1:length(noiseAudio)) / 44100, noiseAudio);
            axis(ax2, 'tight');
        end
    end

    % Function to apply noise cancellation
    function applyNoiseCancellation(~, ~)
        if isempty(originalAudio) || isempty(noiseAudio)
            disp('Please select both original and noise audio files');
            return;
        end

        % Determine the length of the shorter audio file
        minLength = min(length(originalAudio), length(noiseAudio));

        % Pad or truncate arrays to match the length of the shorter audio file
        originalAudio = originalAudio(1:minLength);
        noiseAudio = noiseAudio(1:minLength);

        % Apply noise cancellation
        outputAudio = originalAudio - noiseAudio;

        % Plot noise cancellation audio waveform
```

```
        plot(ax3, (1:length(outputAudio)) / 44100, outputAudio);
        axis(ax3, 'tight');

        % Play the output audio
        sound(outputAudio, 44100);
    end
end
```

CHAPTER 6

RESULTS WITH GRAPH/SIMULATION

Original Audio Waveform:

The first graph displays the waveform of the original audio selected by the user. It shows the amplitude of the audio signal over time.

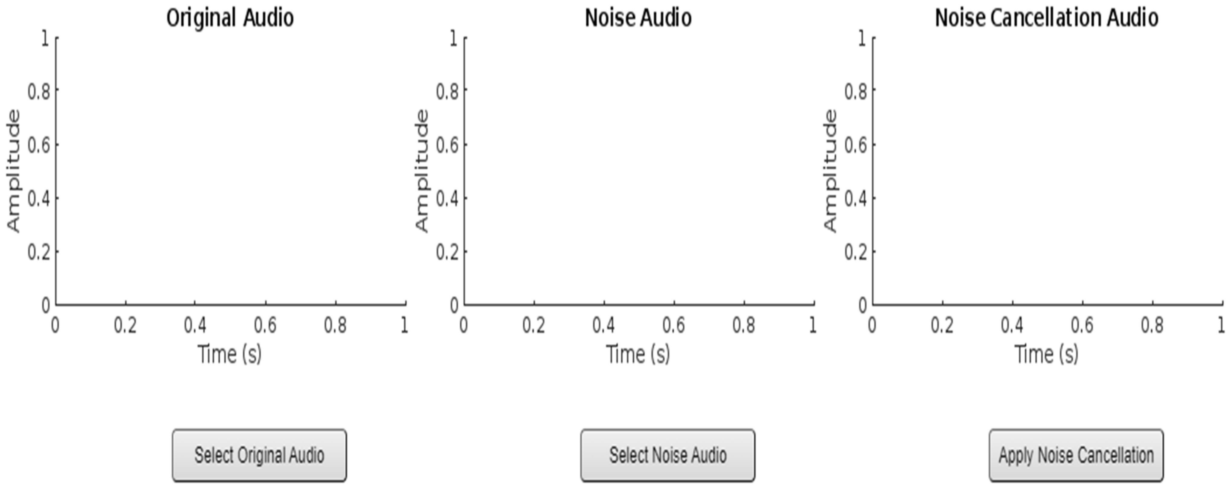
Noise Audio Waveform:

The second graph depicts the waveform of the noise audio selected by the user. It represents the unwanted noise present in the original audio signal.

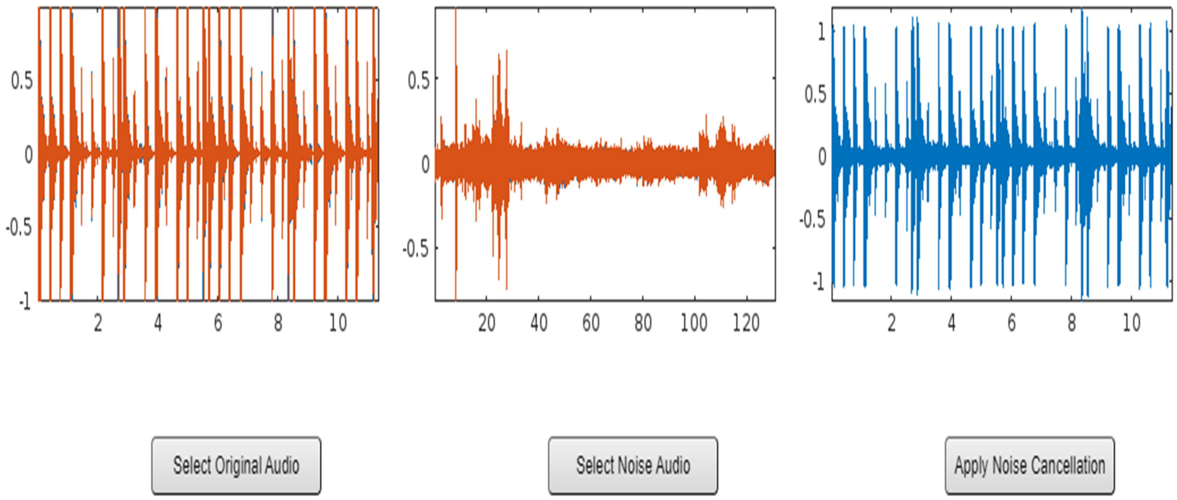
Noise Cancellation Audio Waveform:

The third graph illustrates the waveform of the output audio after noise cancellation has been applied.

Before Loading the input audio files



After Loading the input audio files



CHAPTER 7

INFERENCES

Effectiveness of Noise Cancellation:

The primary inference drawn from the ANC GUI is the effectiveness of the noise cancellation process. By comparing the original audio waveform with the output audio waveform after noise cancellation, users can assess the extent to which unwanted noise has been mitigated. A successful noise cancellation process will result in a noticeable reduction in the amplitude of noise components in the output audio waveform, indicating the effectiveness of the ANC algorithm.

Quality of Audio Signal Preservation:

Another important inference is the quality of audio signal preservation during the noise cancellation process. While reducing unwanted noise is essential, it is equally important to ensure that the desired audio signal remains intact and unaltered. Users can evaluate the quality of audio signal preservation by comparing the output audio waveform with the original audio waveform. Minimal distortion or alteration in the audio signal suggests high-quality noise cancellation with minimal impact on the desired audio content.

Impact of Noise Audio Selection:

The ANC GUI allows users to select noise audio files containing only unwanted noise. The selection of appropriate noise audio files significantly influences the effectiveness of noise cancellation. Inferences can be drawn regarding the impact of different noise audio selections on the quality of noise cancellation. Users can experiment with various noise audio files to understand how different types and levels of noise affect the noise cancellation process and the resulting output audio quality.

Real-time Evaluation and Iterative Improvement:

The real-time evaluation capability of the ANC GUI enables users to iteratively assess and improve the noise cancellation process. By listening to the output audio and visually inspecting the output audio waveform, users can identify areas for improvement and fine-tune parameters or algorithms to enhance noise cancellation performance. This iterative evaluation process fosters continuous improvement and optimization of ANC techniques.

User-Friendly Interaction and Intuitive Visualization:

The ANC GUI provides a user-friendly interface for interacting with audio files and visualizing noise cancellation results. Through intuitive graphical elements, such as pushbuttons and waveform plots, users can seamlessly navigate the application and interpret noise cancellation outcomes. The graphical representation of audio waveforms facilitates clear and insightful visualization of noise reduction and audio signal preservation, enhancing user understanding and engagement.

CHAPTER 8

CONCLUSION

The implementation of the Active Noise Control (ANC) Graphical User Interface (GUI) offers a comprehensive platform for users to delve into the realm of noise cancellation in audio signals. Through interactive features and intuitive visualization tools, users can gain valuable insights and draw meaningful conclusions regarding the effectiveness, quality, and impact of ANC techniques.

The ANC GUI enables users to assess the effectiveness of noise cancellation by comparing the original audio waveform with the output audio waveform. A successful noise cancellation process manifests as a significant reduction in unwanted noise components, affirming the efficacy of the ANC algorithm. Moreover, users can evaluate the preservation of desired audio signals, ensuring minimal distortion or alteration during the noise cancellation process.

The selection of noise audio files plays a crucial role in ANC performance, and the ANC GUI facilitates experimentation with different noise audio selections. Users can observe the influence of various noise types and levels on noise cancellation outcomes, providing valuable insights into noise characteristics and their impact on ANC efficacy.

Furthermore, the real-time evaluation capability of the ANC GUI empowers users to iteratively refine noise cancellation parameters and algorithms. By listening to the output audio and visually inspecting waveform plots, users can identify areas for improvement and optimize ANC techniques for enhanced performance.

In summary, the ANC GUI serves as a versatile tool for exploring, experimenting, and optimizing noise cancellation processes in audio signals. Its user-friendly interface, coupled with intuitive visualization tools, fosters a deeper understanding of ANC principles and facilitates the development of effective noise reduction strategies for diverse applications and scenarios. Through continuous exploration and refinement, the ANC GUI empowers users to unlock the full potential of ANC techniques and elevate audio quality in various environments.