Ahsanullah University of Science & Technology

Department of Electrical and Electronic Engineering

Course Number : *EEE 3208*

Course Title : *Communication Theory LAB*

OEL PROJECT REPORT

Submitted by:

GROUP 5:

ID:

20210205167

20210205170

20210205179

20210205183

3rd Year, 2nd Semester

Section: C
Date: 10/3/25

Audio Processing and Modulation using Matlab App Design

1st MD Reedwan Ahmed
Department of Electrical and Electronics Engineering
Ahsanullah University of Science and Technology,
Dhaka, Bangladesh
Reedwan.eee.20210205167@aust.edu

2nd Mohammed Sami
Department of Electrical and Electronics Engineering
Ahsanullah University of Science and Technology,
Dhaka, Bangladesh
Mohammed.eee.20210205170@aust.edu

3rd Arifin Shohan
Department of Electrical and Electronics Engineering
Ahsanullah University of Science and Technology,
Dhaka, Bangladesh
Arifin.eee.20210205179@aust.edu

4th Miftahul Islam
l*Department of Electrical and Electronics Engineering*
Ahsanullah University of Science and Technology,
Dhaka, Bangladesh
Miftahul.eee.20210205183@aust.edu

***Abstract*—This paper presents the design and implementation of an audio processing system in MATLAB for filtering, modulating, and enhancing the quality of loaded or recorded audio signals. The aim of the project is to split the vocal and noise components, amplitude modulate, and enhance audio quality. The project was created and simulated in MATLAB App Designer with the purpose of allowing users to plot and modify audio signals in the time domain as well as the frequency domain.**

I. INTRODUCTION

A. Purpose and Objectives:

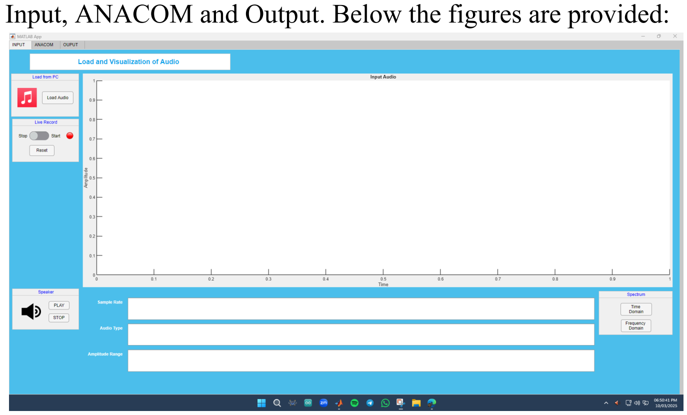
The purpose of the project is to design and implement an audio modulating and processing system with the help of MATLAB. The objectives are:

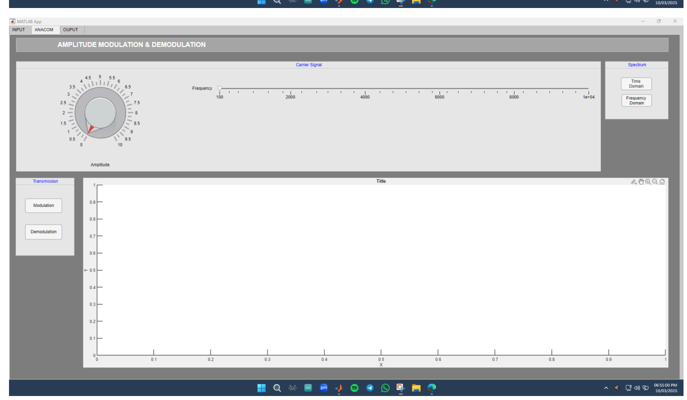
-Designing a voice and noise separation mechanism for an audio signal.
-Carrying out amplitude modulation and demodulation.
-Enabling real-time display of time- and frequency-domain presentations.
-Providing an option for audio enhancement through amplification and filtering.
-Activating audio playback, capture, and save features.

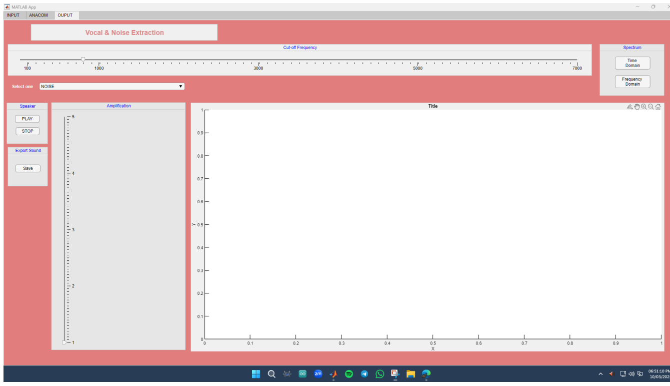
B. Background

Digital communication systems, speech improvement, and signal processing are significant fields of audio processing in the present times. Modulation finds application in radio broadcasting, mobile telephony, and in the field of sound engineering, among others. MATLAB is a versatile digital signal processing tool, and hence it is a very suitable tool for designing and executing such systems.

Our MATLAB project consists of 3 tabs:
Input, ANACOM and Output. Below the figures are provided:







II. EQUIPMENT AND SOFTWARE

A. Software Utilized

-MATLAB R2020a (App Designer for developing a GUI)

-Signal Processing Toolbox

B. Hardware Used

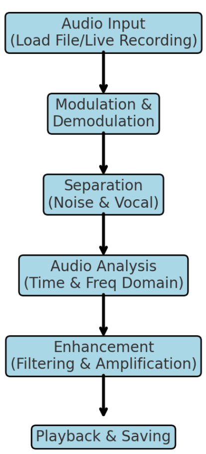
-Personal Computer (Windows)

-Output: External mic (for live record capability)

-Sound playing device (speakers/headphones)

C. Block Diagram

Below is a simplified block diagram representing the system workflow:



III. PROCEDURE

A. Step-by-Step Implementation:

The process is a sequential process:

1. GUI Development

We used the MATLAB App Designer tool for designing an easy to use Graphical User Interface (GUI). The interface features:

-Responsible for audio file handling, record, and saves.

-Managing drop-down selection menus and slider options.

-Axis properties for graph visualization in the time and frequency domains.

-Playback controllers are used in order to hear the processed signal.

2. Audio Loading & Recording

The system allows users to either

-Loading an audio file (audioread()) – WAV, M4A formats are supported.

-The audiorecorder() option offers the ability to capture audio in real time.

-Following the record or load operation of the audio, the audio time domain signal is graphed using plot(). Sample rate, amplitude range, as well as the type of the file, are retrieved -and printed.

3. Time-Frequency Analysis

The recorded or loaded signal is subjected to time and frequency domain analysis:

-Time Domain Visualization – Plots the waveform of the signal with plot().

-Frequency domain analysis utilizes the Fast Fourier Transform operation (fft()) in order to convert the signal into its frequency domain equivalent.

4. Modulation & Demodulation (See Tab ANAC):

The mechanism offers Double Sideband Suppressed Carrier (DSB-SC) Amplitude Modulation, with broad applications in communication systems.

-The carrier signal is generated by cos(2πft).

-The modulated signal depends on the formula (1 + audioData) * carrierSignal.

-The Hilbert transform (hilbert()) is used to obtain the demodulated signal.

-Users can modify the carrier frequency as well as the amplitude using sliders.

5. **Noise & Vocal Separation (Filtering)**

-We apply an 8th-order high-pass Butterworth filter (butter()) to isolate vocals and remove noise. High Frequency Parts → Noise

-Low Frequency Parts → Vocal Content The user can toggle between VOCAL and NOISE graphs by clicking the drop-down menu.

Functions used:

-butter(8, cutoffFreq / (fs/2), 'high') to design the filter.

-filtfilt() to apply zero-phase filtering for better performance.

6. Enhancement Techniques

Audio Enhancement Features The following signal enhancement features are used by the system: Dynamic Range Compression – Reduces loud peaks (use of movmean() for smoothing). Equalization – Employs a band-pass filter (300Hz–3400Hz) for improving voice clarity. Amplification – Users may amplify or attenuate the volume gain using a slider.

7. Playback & Saving

Users may playback (audioplayer()) the processed or original audio. Processed audio may be saved (audiowrite()) in the formats of WAV or M4A files.

B. Code Overview:

-audioread() – Reads audio files

-audiowrite() – Processed audio writing

-f-ft() & ifft() – Fourier Transform for frequency domain visualization

-butter() & filtfilt() – Filtering with a band-pass

-hilbert() – Envelope detection for demodulation

-audioplayer() - Audio Player

IV. RESULTS AND DISCUSSION

A. Data Collection:

They were performed using various audio recordings, which were live recordings and recorded. Obtained results are:

-Sample rate of loaded files and recorded files.

-Time domain and frequency domain descriptions

-Impact of Modulation on Signal Characteristics

B. Analysis of Results:

-Time-Domain Visualization: Provided audio signals in easy-to-view waveform displays.

-Audio pre- and post-processing spectral distribution in the frequency domain.

-Modulation Effect: The AM DSB-SC technique actually modulated audio signals by altering frequency components.

-Noise Reduction Impact: The band-pass filtering effectively separated vocals from the noise.

-Sound clarity was considerably enhanced by the amplification and equalization features.

C. Graphical Results:

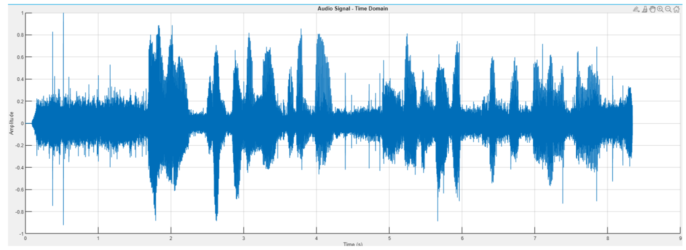


Figure 1: Original Audio Signal - Time Domain

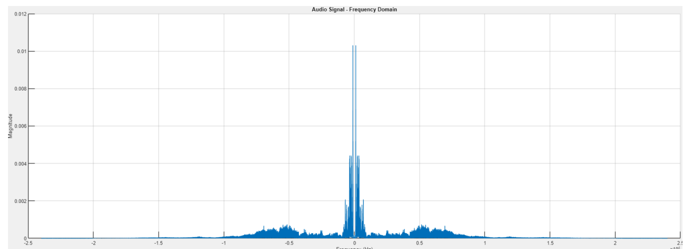


Figure 2: Original Audio Signal - Frequency Domain

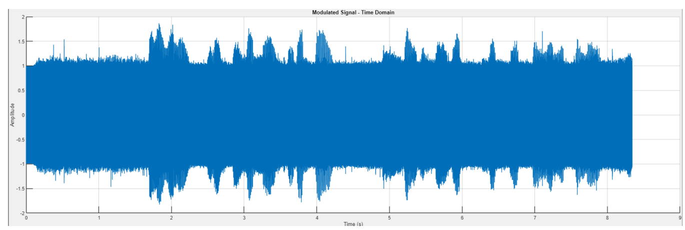


Figure 3: Modulated Signal - Time Domain

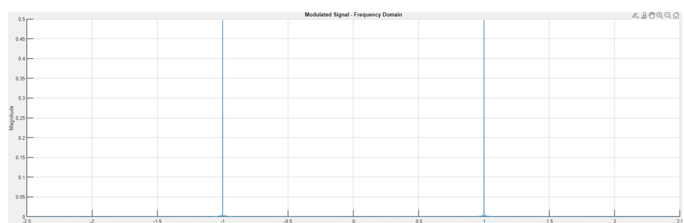


Figure 4: Modulated Signal - Frequency Domain

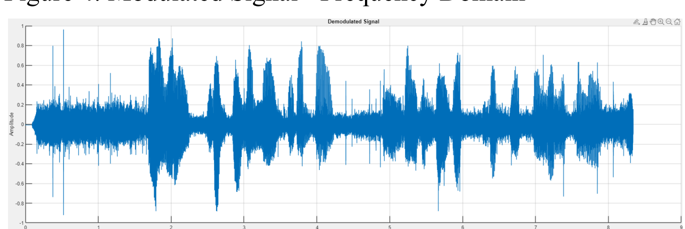


Figure 5: Demodulated Signal - Time Domain

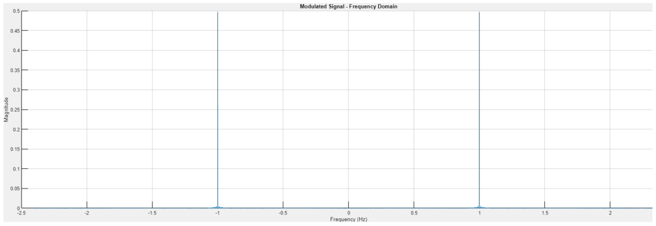


Figure 6: Demodulated Signal - Frequency Domain

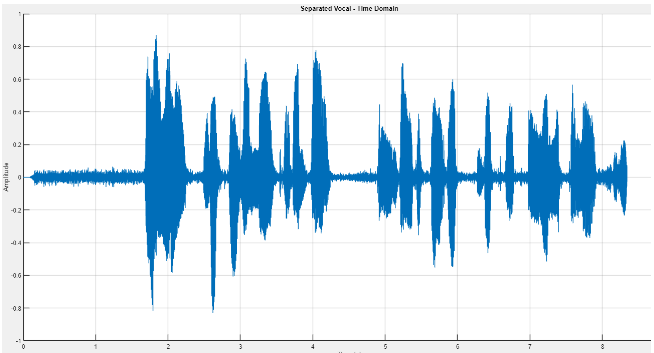


Figure 7: Separated Vocal - Time Domain

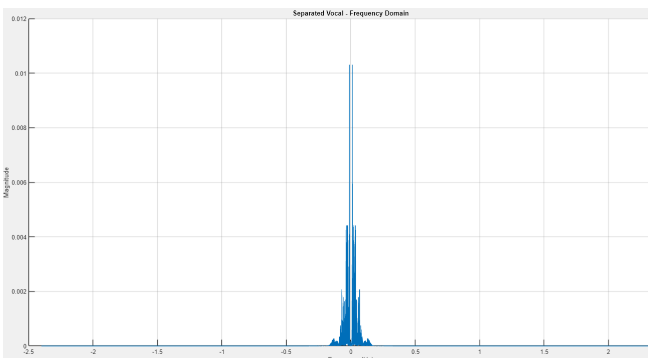


Figure 8: Separated Vocal - Frequency Domain

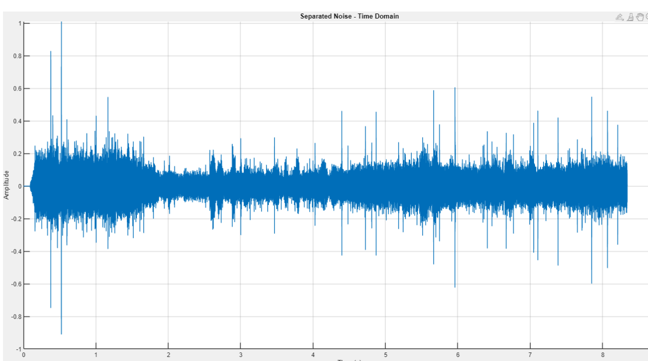


Figure 9: Separated Noise - Time Domain

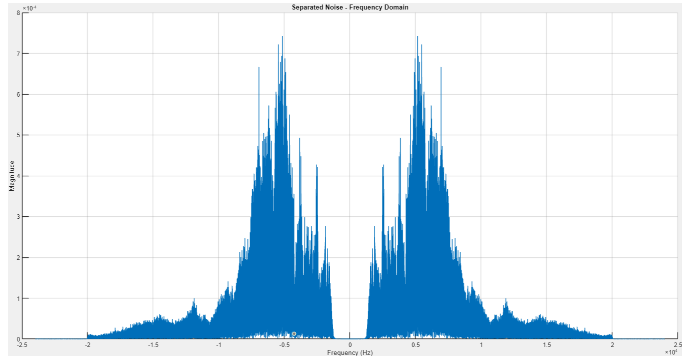


Figure 10: Separated Noise - Frequency Domain

V. CONCLUSION

This project successfully illustrated audio processing and modulation using MATLAB. The system successfully separated vocal and noise components, achieved amplitude modulation, and enhanced sound quality. Real-time visualization assisted in better understanding signal behavior. Some **key observations** are:

-MATLAB is highly suited for digital signal processing and visualization.

-Band-pass filtering is effective in eliminating voice signals in noisy sound.

-AM techniques and demodulation were performed and analyzed effectively.

-The application provides simple interaction to record, process, and save audio.

Limitations:

In this paper, the usage of

-The current setup only captures audio in WAV and M4A (MP3 will require FFmpeg installation).

-Subsequent versions may come with real-time cancellation of noise and machine learning enhancements.

-Hardware implementation using DSP chips or microcontrollers for real-world applications.

References

[1] Oppenheim, A. V., & Schafer, R. W., "Digital Signal Processing," Pearson, 2019.

[2] Proakis, J. G., & Manolakis, D. K., "Digital Signal Processing: Principles, Algorithms, and Applications," Pearson, 2021.
[3] MATLAB Documentation, "Signal Processing Toolbox," MathWorks, 2024.