08 November 2022

Signal Processing Lab Project

Objectives: In this project, we will learn and implement

- Denoising techniques
- Music, instrumental and vocal classifications
- Audio Equalizer
- Filters for real life audio signals
- Hardware Implementation
- **All the test audio files are in the repository - https://github.com/Sasanka-GRS/Signal-Processing-M22-Project**

---

1. Audio Equalizer

Equalization is the process of adjusting the volume of different frequency bands within an audio signal. The circuit or equipment used to achieve this is called an **equalizer**. This can be used in various applications like frequency-based channelling, noise removal, and many more radio applications. The parameters of interest here would include:

1. Number of frequency bands – M
2. A vector of scalars to amplify/attenuate the corresponding frequency band – $y_{M\ x\ 1}$
3. Input audio signal - $x_{N\ x\ 1}$
4. Input audio signal sampling frequency - F_s

The main task in this experiment can be split into sub-tasks:

1. Band-identification: Identify the set of frequency bands for the given number of frequency bands (M)
2. Filtering – Build time-domain IIR/FIR filters for band-passing the bands obtained in previous step
3. Scaling – Scaling each frequency band k, with its corresponding scalar value $y[k]$.

4. Merging – Merge back all the bands to form the final output signal ($x^{eq}_{N\,x\,1}$)

Make sure to explain all the above 4 sub-parts and their importance, theoretically in the report. Note that you are not allowed to do ideal filtering in frequency domain. You are expected to design time domain filters like the ones done in Lab-8.

For this task, the expected specifications include:

1. A MATLAB function equalize(), which takes input the parameters M, y, x, F_s and returns the final equalized signal x^{eq}
2. Write a script to test this function, for the following tasks:
    a. Denoise a noisy signal – Use the given signal ***noisy.wav*** . It is easy to identify what the original signal was, by using the sound() function in MATLAB and listening to it. Your task is to denoise the signal and get as close as possible to the original signal using the equalizer designed
    b. Increase bass – Use the equalizer designed to increase the bass in the given audio file ***bass.mp3***
    c. Self-task – Select any music synthesizing task (for example: treble filter) of your choice and implement it over an audio file of your choice. Expectation from this part deals majorly with the difference between the original audio and the synthesized audio on listening to it and the validity of the filter choice.

2. Classification of pad and bat signals

A **Snickometer** produces an ambient sound or picks up any sound when the ball passes through and then amplifies the appropriate signal. In simple words, **it is a type of microphone that catches sounds of different frequencies that are produced** when the ball hits the bat or any of the leg pads. It helps in the decision-making while there is any **uncertainty in umpires picking up the edges** for caught-behinds or inside-edges in case of LBW decisions. The only parameter of interest is the audio signal obtained, $x_{N\,x\,1}$.

The main task here is to identify whether the given audio signal is of the ball hitting a pad or a bat, using **MATLAB**, post **denoising**.

The specifications here include:

1. Denoising – First task is to denoise the given signals using various denoising techniques. Implement at least 2 denoising techniques (of your choice), as MATLAB functions, with the parameters as input signal and any denoising specific parameters. Explain the denoising techniques theoretically in detail in your reports.
2. Processing and computerized thresholding – Here, identify whether the signal is a pad signal or a bat signal by ***processing*** the signal and thresholding to classify. **HINT :** The obtained time-denoised signal is not enough for classification. Domain change and projection is required. Check the working over the given signals ***1.mp4***, ***2.mp4***, ***1n.mp4*** and ***2n.mp4***.

3. Vocal and Instrument Separation

The only parameter of interest is the audio signal obtained, $x_{N\,x\,1}$.

The main task here is to separate the vocal and instrumental part of a given audio signal.
The specifications of this task are:

1. Identification of varied projection – Identify the primal difference between vocal signals and instrumental signals. This is a major finding of this experiment, and this choice determines the theoretical validation of the separation technique. Justify this choice in your report.
2. Separation – Using simple ***thresholding***, separate the given audio signal into the vocal part $v_{N\,x\,1}$ and the instrumental part $w_{N\,x\,1}$.
3. Synthesis – Use the separated components, process them and combine them to one signal to implement the following tasks over the file ***music.mp3***:
    a. Vocal tone enhancement
    b. Instrumental enhancement
    c. Vocal-instrumental balance
4. What are your processing techniques used for the above 3 tasks? Justify them in detail in the report.

4. Hardware Implementation

Signal Processing deals not just with MATLAB and digital processing, but also deals with the translation of this concept onto hardware and testing the working in real-life situations. The evaluation for this would be done in the Lab during the final lab evaluation.

Implementing in hardware, as discussed in the Lab, will deal with integrating an Arduino with MATLAB, integrating a microphone and a speaker with the Arduino, A/D conversion from microphone input to obtain signal vector processible by MATLAB and D/A conversion of MATLAB output to be played on the speaker. Please note that A/D and D/A conversions can be done directly on the Arduino (that's the main purpose of Arduino here).

The circuit diagram and architectures are left to the students ease of implementation.

This main task here involves 3 subtasks:

1. Implement Task-1 in hardware. Play an audio signal, detect it over a microphone, process it on MATLAB, and hear it over a speaker.
2. Implement Task-2 in hardware. Play an audio, detect it over a microphone, process it on MATLAB, and turn on a GREEN Light for a bat detection or a RED light for a pad detection in an RGB LED.
3. Implement Task-3 in hardware. Play an audio, detect it over a microphone, process it on MATLAB, and hear it over a speaker.
