Voice Classification Projects
Signal and Systems

1 Project Overview

This project involves implementing a system to classify voice signals as either male or female based on their energy and frequency characteristics. The system will use concepts from the Discrete Fourier Transform (DFT), Inverse Discrete Fourier Transform (IDFT), and signal energy analysis.

2 Project Tasks

To complete the project, you will need to:

1. Record and load audio signals.
2. Implement the DFT and IDFT from scratch (with out library).
3. Filter signals in the frequency domain to remove noise.
4. Compute the energy of signals.
5. Classify voice signals as male or female based on energy analysis.
6. Write a report summarizing your implementation and results.

3 Theory and Equations

This section provides the key equations you need to implement for the project.

3.1 Discrete Fourier Transform (DFT)

The DFT transforms a discrete-time signal from the time domain to the frequency domain. It is defined as:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n}.$$

where:

- $x[n]$: The discrete-time signal in the time domain.
- $X(e^{j\omega})$: The Discrete-Time Fourier Transform of $x[n]$, which represents the signal in the frequency domain.

3.2 Inverse Discrete Fourier Transform (IDFT)

The IDFT transforms a frequency-domain signal back to the time domain. It is defined as:

$$x[n] = \frac{1}{2\pi} \int_{2\pi} X(e^{j\omega}) e^{j\omega n} d\omega,$$

3.3 Energy of a Signal

The energy of a signal $x[n]$ is given by:

$$E = \sum_{n=0}^{N-1} |x[n]|^2$$

This equation can be used for signals in either the time domain or the frequency domain.

4 Implementation Steps

Here is the step-by-step guide to complete the project:

4.1 1. Load Audio Signals

Save the audio files as `.wav` files. Use the provided function to load and process these signals.

4.2 2. Implement DFT and IDFT

Write functions to compute the DFT and IDFT using the equations provided. Test your implementation with simple signals such as a sine wave or a delta function.

4.3 3. Filter Signals in the Frequency Domain

Use the DFT to transform signals to the frequency domain. Remove unwanted noise by setting frequency components outside the range of interest (e.g., 50–5000 Hz) to zero. Use the IDFT to transform the filtered signal back to the time domain.

4.4 4. Compute Signal Energy

Write a function to compute the energy of a signal using the formula provided. Compare the energy of filtered signals with predefined reference values.

4.5 5. Classify Voice Signals

After processing the provided files, your system should be capable of recording a voice and instantly processing it to determine if it is a male or female voice.

Compare the energy of a new voice signal with the average energies of male and female signals. Classify the voice as "Male" or "Female".

5 Loading .wav Files and Extracting Discrete Signals

In order to process the voice signals in your project, you need to first load the audio data from '.wav' files and convert them into discrete signals that you can work with. The following Python function is provided to help you achieve this:

```
def get_discrete_signal(file_path, resample_rate=None):
    with wave.open(file_path, 'r') as wav_file:
        original_sample_rate = wav_file.getframerate()
        n_frames = wav_file.getnframes()
        duration = n_frames / original_sample_rate
        n_channels = wav_file.getnchannels()

        raw_data = wav_file.readframes(n_frames)
        signal = np.frombuffer(raw_data, dtype=np.int16)

        if n_channels > 1:
            signal = signal[::n_channels]

        time_values = np.linspace(0, duration, len(signal), endpoint=False)

        if resample_rate and resample_rate != original_sample_rate:
            from scipy.signal import resample
            num_samples = int(duration * resample_rate)
            signal = resample(signal, num_samples)
            time_values = np.linspace(0, duration, len(signal), endpoint=False)

        return signal, time_values, original_sample_rate
```

6 Report Deliverables

Your final submission should include:

- Your code implementing all the tasks.
- A report discussing:
  - Your implementation of DFT, IDFT, and signal energy computation.
  - Plot the input signal before and after applying a filter, and also show the results of the Discrete Fourier Transform (DFT) when the signal is in the frequency domain.