RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Digital Signal Processing Assignment

Convolution of Audio Signals with Digital Filters

Submitted By:

MD. Mehedi Hasan Maruf
Roll: 2003037
Department of CSE
RUET, Rajshahi

Submitted To:

Md. Farukuzzaman Faruk
Assistant Professor
Department of CSE
RUET, Rajshahi

Date of Submission: May 26, 2025

Contents

1 Introduction

DSP has become an essential branch in modern communication and audio system design. This work deals with the practical implementation of convolution operations in audio signals using digital filters. Convolution is the prime mathematical operation that relates input signals, system response, and output signals in linear time-invariant (LTI) systems. In the present study, we are actualizing and analyzing the convolution of an actual audio signal with a band-pass filter in order to demonstrate frequency-selective filtering. This project employs Python and scientific libraries for computing to process, analyze, and visualize the effects of digital filtering on audio signals. The primary aim is to understand how the convolution operations have been utilized to modify the spectral attribute of sound signals so that unwanted frequencies are removed from the frequency content while ceiling desired frequency ranges.

2 Motivation

The motivation for this assignment stems from the widespread applications of audio signal processing in various domains:

- **Audio Enhancement**: Improving audio quality by removing noise and unwanted frequencies
- **Communication Systems**: Filtering signals to meet transmission bandwidth requirements
- **Music Production**: Creating special effects and sound modification
- **Hearing Aids**: Amplifying specific frequency ranges to assist hearing-impaired individuals
- **Speech Processing**: Extracting relevant speech features for recognition systems

Understanding convolution operations is essential for anyone working in audio engineering, telecommunications, or multimedia systems. This practical implementation bridges the gap between theoretical knowledge and real-world applications.

3 Background Study

3.1 Digital Signal Processing Fundamentals

Digital Signal Processing involves the manipulation of signals using computational algorithms. Key concepts include:

- **Sampling**: Converting continuous-time signals to discrete-time signals
- **Quantization**: Converting continuous amplitude values to discrete levels
- **Digital Filters**: Systems that modify the frequency content of signals
- **Fourier Transform**: Tool for analyzing frequency content of signals

3.2 Audio Signal Characteristics

Audio signals typically have the following characteristics:

- Frequency range: 20 Hz to 20 kHz (human audible range)
- Common sampling rates: 44.1 kHz (CD quality), 48 kHz (professional audio)
- Dynamic range: Often represented as 16-bit or 24-bit samples

3.3 Filter Types

Common digital filter types include:

- **Low-pass**: Allows low frequencies, attenuates high frequencies
- **High-pass**: Allows high frequencies, attenuates low frequencies
- **Band-pass**: Allows frequencies within a specific range
- **Band-stop**: Attenuates frequencies within a specific range

4 Social and Economic Impact

4.1 Social Impact

The application of audio signal processing through convolution has significant social implications:

- **Accessibility**: Hearing aid technology improves quality of life for millions of people with hearing impairments
- **Entertainment Industry**: Enhanced audio quality in music, films, and broadcasting
- **Communication**: Improved voice quality in telecommunication systems
- **Education**: Better audio systems in classrooms and online learning platforms
- **Safety**: Enhanced emergency communication systems and noise monitoring

4.2 Economic Impact

The economic significance of audio signal processing is substantial:

- **Market Size**: The global audio DSP market is valued at several billion dollars
- **Job Creation**: Thousands of jobs in audio engineering, software development, and research
- **Innovation**: Drives technological advancement in consumer electronics

- **Cost Reduction**: Digital filters are more cost-effective than analog alternatives
- **Industry Growth**: Supports growth in telecommunications, entertainment, and healthcare sectors

5 Related Mathematical Studies

5.1 Convolution Theory

The mathematical foundation of this assignment is based on convolution theory. For discrete-time signals, convolution is defined as:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \tag{1}$$

where:

- $y[n]$ is the output signal
- $x[n]$ is the input signal
- $h[n]$ is the impulse response of the filter
- $*$ denotes the convolution operation

5.2 Frequency Domain Analysis

The Fourier Transform relationship shows that convolution in time domain corresponds to multiplication in frequency domain:

$$Y(\omega) = X(\omega) \cdot H(\omega) \tag{2}$$

where $Y(\omega)$, $X(\omega)$, and $H(\omega)$ are the Fourier transforms of $y[n]$, $x[n]$, and $h[n]$ respectively.

5.3 Z-Transform

For digital systems, the Z-transform provides a powerful tool for analysis:

$$H(z) = \frac{Y(z)}{X(z)} \tag{3}$$

where $H(z)$ is the transfer function of the system.

6 Theory

6.1 Linear Time-Invariant Systems

LTI systems are characterized by two important properties:

- **Linearity**: $T\{a_1 x_1[n] + a_2 x_2[n]\} = a_1 T\{x_1[n]\} + a_2 T\{x_2[n]\}$
- **Time-Invariance**: If $y[n] = T\{x[n]\}$, then $y[n - n_0] = T\{x[n - n_0]\}$

6.2 Butterworth Filter Design

The band-pass filter used in this implementation is based on the Butterworth design, characterized by:

$$|H(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}} \tag{4}$$

where:

- N is the filter order
- ω_c is the cutoff frequency
- The response is maximally flat in the passband

6.3 Band-pass Filter Implementation

A band-pass filter can be implemented by cascading high-pass and low-pass filters or by using direct design methods. The transfer function for a band-pass filter is:

$$H(z) = \frac{\sum_{i=0}^{M} b_i z^{-i}}{1 + \sum_{j=1}^{N} a_j z^{-j}} \tag{5}$$

where b_i and a_j are the filter coefficients.

7 Implementation

7.1 Software and Libraries

The implementation utilizes Python with the following libraries:

- **NumPy**: Numerical computing and array operations
- **SciPy**: Scientific computing, specifically `scipy.signal` for filter design
- **Matplotlib**: Data visualization and plotting
- **IPython**: Audio playback capabilities

7.2 System Architecture

The implementation follows a modular approach:

1. Audio loading and preprocessing
2. Filter design and coefficient calculation
3. Convolution operation (filtering)
4. Analysis and visualization
5. Audio playback and comparison

7.3 Filter Specifications

The band-pass filter implemented has the following specifications:

- Lower cutoff frequency: 500 Hz
- Upper cutoff frequency: 3000 Hz
- Filter order: 5
- Filter type: Butterworth
- Implementation: IIR (Infinite Impulse Response)

8 Code Segments

8.1 Audio Loading and Preprocessing

```python
# Load audio from URL
audio_url = "https://github.com/mehedi37/DSP_LAB/raw/refs/heads/main/W3KM.wav"
response = requests.get(audio_url)
audio_bytes = BytesIO(response.content)
sample_rate, audio_data = wavfile.read(audio_bytes)

# Convert to mono and normalize
if len(audio_data.shape) > 1:
    audio_data = audio_data[:, 0]
if audio_data.dtype == np.int16:
    audio_data = audio_data.astype(np.float32) / 32767.0
```

Listing 1: Audio Loading from URL

8.2 Filter Design

```python
def create_bandpass_filter(lowcut, highcut, sample_rate, filter_order=5):
    """Create a band-pass filter using scipy.signal"""
    nyquist = 0.5 * sample_rate
    low = lowcut / nyquist
    high = highcut / nyquist
    b, a = signal.butter(filter_order, [low, high], btype='band')
    return b, a

# Create filter coefficients
bandpass_b, bandpass_a = create_bandpass_filter(
    lowcut=500, highcut=3000, sample_rate=sample_rate
)
```

Listing 2: Band-pass Filter Design

8.3 Convolution Operation

```python
# Apply filter using convolution
filtered_bandpass = signal.lfilter(bandpass_b, bandpass_a, audio_data)
```

Listing 3: Filter Application

8.4 Visualization Functions

```python
def plot_waveform(signal_data, sr, title="Waveform", alpha=1.0, color='blue'):
    """Plot the waveform of an audio signal"""
    time = np.arange(0, len(signal_data)) / sr
    plt.plot(time, signal_data, alpha=alpha, color=color, label=title)
    plt.grid(True)
    plt.xlabel('Time (seconds)')
    plt.ylabel('Amplitude')
    plt.title(title)

def plot_spectrum(signal_data, sr, title="Frequency Spectrum", alpha=1.0, color='blue'):
    """Plot the frequency spectrum of an audio signal"""
    n = len(signal_data)
    yf = np.fft.rfft(signal_data)
    xf = np.fft.rfftfreq(n, 1 / sr)
    magnitude = 20 * np.log10(np.abs(yf) / n + 1e-10)
    plt.semilogx(xf, magnitude, alpha=alpha, color=color, label=title)
    plt.grid(True, which="both")
```

```
    plt.xlabel('Frequency (Hz)')
    plt.ylabel('Magnitude (dB)')
    plt.title(title)
    plt.xlim(20, sr/2)
```

Listing 4: Waveform and Spectrum Plotting

9 Output and Results

9.1 Audio Signal Analysis

The implementation successfully processed the audio signal with the following characteristics:

- Sample rate: Variable (depends on input file)
- Duration: Truncated to 10 seconds for processing efficiency
- Channels: Converted to mono for simplicity
- Data type: Normalized floating-point representation

9.2 Filter Performance

The band-pass filter (500 Hz - 3 kHz) effectively:

- Attenuated low-frequency components below 500 Hz
- Preserved mid-frequency content (500 Hz - 3 kHz)
- Attenuated high-frequency components above 3 kHz
- Maintained the overall temporal structure of the signal

9.3 Visual Analysis

The visualization results demonstrate:

- Clear differences between original and filtered waveforms
- Frequency spectrum showing the band-pass characteristics
- Overlay comparisons highlighting the filtering effects
- Time-domain analysis showing amplitude changes

9.4 Auditory Results

The audio playback functionality allows for:

- Direct comparison between original and filtered audio
- Subjective evaluation of filtering quality
- Understanding of perceptual effects of filtering

10 Conclusion

This assignment successfully demonstrated the practical application of convolution operations in audio signal processing. The key achievements include:

10.1 Technical Accomplishments

- Successful implementation of digital band-pass filtering
- Effective use of Python scientific computing libraries
- Comprehensive visualization of time and frequency domain effects
- Integration of audio playback for subjective evaluation

10.2 Learning Outcomes

- Understanding of convolution theory and its practical applications
- Experience with digital filter design and implementation
- Knowledge of audio signal characteristics and processing techniques
- Proficiency in scientific computing and data visualization

10.3 Future Work

Potential extensions of this work include:

- Implementation of different filter types (FIR vs. IIR comparison)
- Real-time audio processing applications
- Adaptive filtering techniques
- Multi-band processing and equalization
- Noise reduction and enhancement algorithms

10.4 Final Remarks

The application of the convolution of audio signals through digital filters represents the most fundamental operation in the digital signal analysis approach. This gives the student a practical experience of the theoretical foundations and practical implications in digital audio signal processing. The analysis of the set of results presents evidence for the capacity and versatility of digital filtering procedures and corporate-side intervention to change the properties of signals without compromise on computational efficiency. Knowledge gained under this assignment is transferable to different domains, such as audio engineering, telecommunications, multimedia systems, and biomedical signal processing. As new advanced schemes continue to be evolved for digital signal processing implementations, an understanding of basic signal processing operations still remains indispensable for implementing new audio and communication techniques.