Chapter 5. Digital Audio Processing
Objectives

• Understand how normalization, compression, expansion, equalization, and reverb are applied and what they do to digital audio.
• Understand methods for audio restoration.
• Understand how filters are applied and how they work mathematically.
• Understand the concept and examples of time-based encoding for digital audio.
• Understand the concept and examples of perceptual encoding for digital audio.
• Understand the concept, implementation, and application of MPEG audio compression.
DBFS

- In audio processing softwares, the amplitude is often shown in **dBFS. (decibels-full-scale)**

  \[ \text{dBFS} = 20 \log_{10} \left( \frac{|x|}{2^{n-1}} \right) \]

- The center of the amplitude axis is \(-\infty\) dBFS, and above and below this axis the values progress to the maximum of 0 dBFS.
Dynamics Processing

- **Dynamics processing** is the process of adjusting the dynamic range of an audio selection, either to reduce or to increase.

- An increase in amplitude is called **gain** or **boost**. A decrease in amplitude is called **attenuation** or, informally, a **cut**.

- We introduce 4 digital dynamics processing tools here: **hard limiting**, **normalization**, **compression**, and **expansion**.
Types of dynamic range compression and expansion
Compression and Expansion

- **Downward compression** lowers the amplitude of signals that are above a designated level, without changing the amplitude of signals below the designated level. It reduces the dynamic range.

- **Upward compression** raises the amplitude of signals that are below a designated level without altering the amplitude of signals above the designated level. It reduces the dynamic range.

- **Upward expansion** raises the amplitude of signals that are above a designated level, without changing the amplitude of signals below that level. It increases the dynamic range.

- **Downward expansion** lowers the amplitude of signals that are below a designated level without changing the amplitude of signals above this level. It increases the dynamic range.
Compression and Expansion - Examples

- Downward compression:
- Amplitudes above -40dB is lowered by a 2 : 1 ratio.
Compression and Expansion - Examples

- Upward compression:
- Amplitudes below -30dB is compressed by a 2 : 1 ratio.
Limiting

- **Audio limiting** limits the amplitude of an audio signal to a designated level.
- **Hard limiting (clipping)**
  - cuts amplitudes of samples to a given maximum and/or minimum level.
- **Soft limiting**
  - audio signals above the designated amplitude are recorded at lower amplitude.
Limiting

Original Signal

Hard Clipping (Limiting with zero attack and release)

Soft Clipping

http://en.wikipedia.org/wiki/File:Clipping_compared_to_limiting.svg
Normalization

- Often, normalization is used to increase the perceived loudness of a piece after the dynamic range of the piece has been compressed.
- Normalization steps:
  1. Find the highest amplitude sample in the audio selection.
  2. Determine the gain needed in the amplitude to raise the highest amplitude to maximum amplitude.
  3. Raise all samples in the selection by this amount.
Dynamics Processing - Example

- Bossa.wav - original

- Bossa.wav - normalized
Dynamics Processing - Example

- Bossa.wav - compressed

- Bossa.wav – compressed + normalized
Audio Restoration

• In this section, we introduce three basic types of audio restoration to alleviate the background noise that arises from the microphone, air, disk ...etc.
  • Noise gating
  • Noise reduction
A noise gate allows a signal to pass through only when it is above a set threshold. It is used when the level of the signal is above the level of the noise. It does not remove noise from the signal. When the gate is open, both the signal and the noise will pass through.

http://en.wikipedia.org/wiki/Noise_gate
Noise Gating

- **Reduction Level**: the amplitude to which you want the below-threshold samples to be reduced.
- **Attack**: the attack time indicates how quickly you want the gate to open when the signal goes above the threshold, like fade-in.
- **Release**: The release time indicates how quickly you want the gate to close, like fade-out.
- **Hold**: the amount of time the gate will stay open after the signal falls below the threshold.
Noise Gating

- If the signal keeps moving back and forth around the threshold, the gate will open and close continuously, creating a kind of chatter.

- The hysteresis control indicates the difference between the value $n$ that caused the gate to open and the value $m$ that will cause it to close again. If $n - m$ is large enough to contain the fluctuating signal, the noise gate won’t cause chatter.
Noise Reduction

- Steps for noise reduction:
  1. Get a profile of the background noise. This can be done by selecting an area that should be silent, but that contains a hum or buzz.
  2. Determine the frequencies in the noise and their corresponding amplitude levels.
  3. The entire signal is processed in sections (FFT). The frequencies in each section are analyzed and compared to the profile, and if these sections contain frequency components similar to the noise, they can be eliminated below certain amplitudes.
Noise Reduction - Example

- Reduce the noise of ‘Bossa_dithered.wav’

The area that should be silent
Noise Reduction - Example

- Before noise reduction

- After noise reduction
A digital audio filter is a linear system that changes the amplitude or phase of one or more frequency components of an audio signal.

Types of digital audio filter
- FIR (finite-impulse response) filter
- IIR (infinite-impulse response) filter
Let $x(n)$ be an audio signal of $L$ samples for $0 \leq n \leq L - 1$

$y(n)$ be the filtered signal

$h(n)$ be the convolution mask

- The FIR filter function is defined by

$$y(n) = h(n) \otimes x(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

where $x(n-k) = 0$ if $n-k < 0$

- The number of the coefficients of $h$ is the order of the filter.
The infinite form of the IIR filter function is defined by

\[ y(n) = h(n) \otimes x(n) = \sum_{k=0}^{\infty} h(k)x(n - k) \]

where \( x(n - k) = 0 \) if \( n - k < 0 \)

However, finding the values \( h(n) \) for an infinitely long mask is impossible. The equation can be transformed to a more manageable difference equation form.
IIR Filter

- The recursive form of IIR filter is

\[ y(n) = h(n) \otimes x(n) = \sum_{k=0}^{N-1} a_k x(n - k) - \sum_{k=1}^{M} b_k y(n - k) \]

- \( a_k \) is the coefficient of the forward filter
- \( b_k \) is the coefficient of the feedback filter
- \( y(n) \) depends on present and past input samples as well as on past outputs.
Impulse and Frequency Response

• The convolution mask $h(n)$ for an FIR or IIR filter is sometimes referred to as the **impulse response**.
• The frequency response, $H(z)$, represents $h(n)$ in frequency domain.
• A **frequency response graph** describes how a filter acts on an audio signal.
Filters in Audio Processing

- **Band filters**
  - **low-pass filter**—retains only frequencies below a given level.
  - **high-pass filter**—retains only frequencies above a given level.
  - **bandpass filter**—retains only frequencies within a given band.
  - **bandstop filter**—eliminates all frequencies within a given band.
Filters in Audio Processing

• **Comb filter** — add delayed versions of a wave to itself, resulting in phase cancellations that can be perceived as echo. Phase cancellations eliminate frequency components when two sine waves that are out-of-phase with each other are summed. Thus, the frequency response has the shape of a comb.

![Frequency response of a comb filter](image)
Filters in Audio Processing

- **Shelving filters** — shelving filters are similar to low- and high-pass filters except that they boost or cut frequencies up to a certain frequency.

![Diagrams of shelving filters showing boosting and cutting effects](Diagrams)
Filters in Audio Processing

- **Peaking filter** — Ideally, for a bandpass filter, the unwanted frequencies would be filtered out entirely, but in reality this is not possible. The frequency response looks more like the bell curve. This type of filter is sometimes called a peaking filter.

- Given the graph of a peaking filter, let $f_{width}$ be the width of the peak measured at a point that is $\frac{1}{2}$ times the peak’s height, and let $f_{center}$ be the frequency at the geometric center of the peak, both in Hz. Then the Q-factor, Q, is defined as

\[
Q = \frac{f_{center}}{f_{width}}
\]
Relationship Between Convolution and Fourier Transform

- Let $H(z)$ be the discrete Fourier transform of a convolution filter $h(n)$, and let $X(z)$ be the discrete Fourier transform of a digital audio signal $x(n)$. Then $y(n) = h(n) \ast x(n)$ is equivalent to the inverse discrete Fourier transform of $Y(z)$, where $Y(z) = H(z)X(z)$. 

Equivalent operations in time and frequency domains
FIR Filter Design – Terms

- The convolution mask $h(n)$ is also called the impulse response, representing a filter in the time domain.
- Its counterpart in the frequency domain, the frequency response $H(z)$, is also sometimes referred to as the transfer function.
- A frequency response graph can be used to show the desired frequency response of a filter you are designing.
FIR Filter Design – Ideal Filter

- An ideal low-pass frequency response graph, normalized.
- In the graph, angular frequency is on the horizontal axis. The vertical axis represents the fraction of each frequency component to be permitted in the filtered signal.
- The cutoff frequency $\omega_c$ must be less than $\pi$ (Nyquist theorem).

\[ \omega_c = 2\pi \frac{f_c}{f_{\text{sample}}} \]

Figure 5.42
FIR Filter Design – Ideal Filter

- The frequency response graph shown in Figure 5.42 is an example of a **rectangle function**.
- If it is an idealized form of $H(z)$, then what would be the corresponding ideal impulse response, an idealized $h(n)$?
- The inverse Fourier transform of a rectangle function in the frequency domain is a **sinc function** in the time domain (Derivation on textbook p.295)

$$h_{\text{ideal}}(n) = \frac{\sin(2\pi f_c n)}{\pi n} \quad \text{for } -\infty \leq n \leq \infty, n \neq 0$$

and $2f_c$ for $n = 0$
**FIR Filter Design – Ideal Filter**

<table>
<thead>
<tr>
<th>Type of filter</th>
<th>Equation for $h_{\text{ideal}}(n)$, $n \neq 0$</th>
<th>$h_{\text{ideal}}(0)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-pass</td>
<td>$\frac{\sin(2\pi f_c n)}{\pi n}$</td>
<td>$2f_c$</td>
</tr>
<tr>
<td>High-pass</td>
<td>$\frac{\sin(2\pi f_c n)}{\pi n}$</td>
<td>$1 - 2f_c$</td>
</tr>
<tr>
<td>Bandpass</td>
<td>$f_2 \frac{\sin(2\pi f_{2n})}{\pi n} - f_1 \frac{\sin(2\pi f_{1n})}{\pi n}$</td>
<td>$2(f_2 - f_1)$</td>
</tr>
<tr>
<td>Bandstop</td>
<td>$f_1 \frac{\sin(2\pi f_{1n})}{\pi n} - f_2 \frac{\sin(2\pi f_{2n})}{\pi n}$</td>
<td>$1 - 2(f_2 - f_1)$</td>
</tr>
</tbody>
</table>
FIR Filter Design – Ideal to Real

• A sinc function goes on infinitely in the positive and negative directions ... how can we implement an FIR filter?
• One way is to multiply the ideal impulse response by a windowing function. The purpose of the windowing function is to make the impulse response finite.
• However, making the impulse response finite results in a frequency response that is less than ideal, containing “ripples”.
The **passband** corresponds to the frequencies the filter tries to retain. The **stopband** corresponds to the frequencies the filter attenuates or filters out.

In a real filter, a **transition band** lies between passband and stopband, and the slope is not infinitely steep, as in an ideal filter.

![Frequency response of a realistic low-pass filter](image)
FIR Filter Design – Windowing Function

• Rectangular windowing function
  • $1$

• Hanning windowing function
  • $w(n) = 0.54 + 0.46\cos\left(\frac{2\pi n}{N}\right)$

• Hamming windowing function
  • $w(n) = 0.5 + 0.5\cos\left(\frac{2\pi n}{N}\right)$

• Blackman windowing function
  • $w(n) = 0.42 + 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right)$
FIR Filter Design - Algorithm

/*Input: f_c, the cutoff frequency for the lowpass filter, in Hz 
   f_samp, the sampling frequency of the audio signal to be filtered, in Hz 
   N, the order of the filter; assume N is odd 
Output: a low-pass FIR filter in the form of an N-element array */

/*Normalize f_c and ω_c so that π is equal to the Nyquist angular frequency*/
f_c = f_c/f_samp
ω_c = 2*π*f_c
middle = N/2 /*Integer division, dropping remainder*/
/*Create the filter using the low-pass filter function from Table 5.2*/
/*Put a dummy value in for n = 0 to avoid a divide by 0 error*/
for n = −N/2 to N/2
   if (n = 0) fltr(middle) = 1
   else fltr(n + middle) = sin(2*π*f_c*n)/(π*n)
fltr(middle) = 2*f_c
/*Multiply the elements of fltr by a windowing function chosen from Table 5.3. We use the Hanning window function*/
for n = 0 to N-1
   fltr(n) = fltr(n) * (0.5 + 0.5*cos((2*π*n)/N))
Digital Audio Compression

• Time-Based Compression Methods
  • No need to transform the data into frequency domain.
  • A-law encoding, $\mu$-law encoding (Ch4) … etc.
  • These methods are often considered conversion techniques rather than compression methods.

• The most effective audio compression methods require some information about the frequency spectrum of the audio signal. These compression methods are based on psychoacoustical modeling and perceptual encoding.
Psychoacoustics

- Psychoacoustics (心理聲學)
  - The study of subjective human perception of sounds.
  - The study of all the psychological interactions between humans and the world of sound.
- Psychoacoustical tests have shown that there is a great deal of nonlinearity in human sound perception.
- The octave from middle C (called C4) to C5 ranges from 261.63 Hz to 523.25 Hz, while C5 to C6 ranges from 523.25 to 1046.50 Hz. But the distance from C4 to C5 subjectively sounds the same as the distance from C5 to C6.
Psychoacoustics

- Humans hear best in the 1000 to 5000 Hz range, the frequency range of human speech.
- For a 100Hz and 1000Hz tone, the 100Hz tone needs larger amplitude to sound equally loud as the 1000Hz.
- Similarly, for a 10000Hz and 1000Hz tone, the 10000Hz tone needs larger amplitude to sound equally loud as the 1000Hz.
Psychoacoustics

- At low frequencies, we can distinguish between sound waves that are only a few Hz apart. At high frequencies, the frequencies must be a hundred or more Hz apart for us to hear the difference. The reason is that the ear is divided into frequency bands called *critical bands*.
- Human ear perceive different frequencies by the response of different critical bands.
- Critical bands are narrower for low-frequency than for high-frequency sounds. Between 1 and 500 Hz, bands are about 100 Hz in width. The critical band at the highest audible frequency is over 4000 Hz wide.
Psychoacoustics

- **Masking** - Within a small window of time, the loudest frequency sound can overpower the others in the critical band, making human unable to hear the other frequencies.
A Sketch of Compression

- A small window of time called a **frame** is moved across a sound file.
- In each frame, use different filters to divide the frame into bands of different frequencies.
- Calculate the masking curve for each band.
- Requantize the samples using fewer bits such that the quantization error is below the masking curve. That is, make sure the noise is inaudible.
Overview of MPEG

- Acronym for the *Motion Picture Experts Group*, a family of compression algorithms for both digital audio and video.
- MPEG has been developed in a number of phases: MPEG-1, MPEG-2, MPEG-4, MPEG-7.
- MPEG-1 covers CD-quality audio suitable for video games.
- MPEG audio is also divided into three layers: Audio Layers I, II, and III.
- Each higher layer is more computationally complex, and generally more efficient at lower bitrates than the previous.
- The well-known MP3 audio file format is actually MPEG-1 Audio Layer III.
MPEG-1 Audio Compression

- Basic concepts of MPEG-1 compression
MPEG-1 Audio Compression - Step 1

- Divide the audio file into frames and analyze the psychoacoustical properties of each frame individually.
  - To analyze the masking phenomenon, we must look at a small piece of time because it happens when different frequencies are played at close to the same time.
  - Frames can contain 384, 576, or 1152 samples, depending on the MPEG phase and layer.
  - For the remainder of these steps, it is assumed that we’re operating on an individual frame.
• By applying a bank of filters, separate the signal into frequency bands.
  • The samples are divided into frequency bands for psychoacoustical analysis. Each filter removes all frequencies except for those in its designated band.
  • The use of filter banks is called subband coding. In MPEG-1, the number of filters is 32.
  • In MPEG-1 Layers I and II, the frequency bands created by the filter banks are uniform in size, which doesn’t match the width of the critical bands in human hearing. Layer III models critical bands more closely than layer I and II.
MPEG-1 Audio Compression - Step 3

- Perform a Fourier transform on the samples in each band in order to analyze the band’s frequency spectrum.
  - After the Fourier transform, we can know exactly how much of each frequency component occurs in each band.
  - From the frequency spectrum, a masking curve can be produced for each band.
• Analyze the influence of tonal and nontonal elements in each band. (Tonal elements are simple sinusoidal components, such as frequencies related to melodic and harmonic music. Nontonal elements are transients like the strike of a drum or the clapping of hands.)
• The longer the time window is in frequency analysis, the better the frequency resolution, but the worse the time resolution. This in turn can mean that transient signals may not be properly identified. Therefore we’d better identify the transient first.
MPEG-1 Audio Compression - Step 5

- Determine how much each band’s influence is likely to spread to neighboring frequency bands.
  - It isn’t sufficient to deal with bands entirely in isolation from each other, since there can be a masking effect between bands.
Find the masking threshold and signal-to-mask ratio (SMR) for each band, and determine the bit depth of each band accordingly.

SMR: the ratio between the peak sound pressure level and the masking threshold.
The masking phenomenon causes the noise floor within a band to be raised. When the noise floor is high relative to the maximum sound pressure level within a band, then fewer bits are needed.

Fewer bits create more quantization noise, but it doesn’t matter if that quantization noise is below the masking threshold. The noise won’t be heard anyway.
MPEG-1 Audio Compression - Step 7

• Quantize the samples for the band with the appropriate number of bits, possibly following this with Huffman encoding.
  • MPEG-1 Layers 1 and 2 use linear quantization, while MP3 uses nonlinear.
algorithm MPEG-1 audio
/*Input: An audio file in the time domain
Output: The same audio file, compressed*/
{
Divide the audio file into frames
For each frame {
    By applying a bank of filters, separate the signal into frequency bands.
    For each frequency band {
        Perform a Fourier transform to analyze the band’s frequency spectrum
        Analyze the influence of tonal and nontonal elements (i.e., transients)
        Analyze how much the frequency band is influenced by neighboring bands
        Find the masking threshold and signal-to-mask ratio (SMR) for the band,
        and determine the bit depth in the band accordingly
        Quantize the samples from the band using the determined bit depth
        Apply Huffman encoding (optional)
    }
Create a frame with a header and encoded samples from all bands
}
Summary

- Dynamic processing
- Audio restoration
- Digital audio filters
- FIR filter design
- Perceptual encoding
- MPEG-1 compression