MMC 2

Thursday, May 8, 2025 1:24 PM

Unit 2: Sound/Audio System

Syllabus Topics:

1. **Concepts of Sound System**
    - Frequency
    - Amplitude
    - Computer representation of sound
    - Sampling rate
    - Quantization
    - Sound hardware
2. **Music and Speech**
    - Basic MIDI concepts
    - MIDI devices
    - MIDI messages
    - MIDI and SMPTE timing standards
    - MIDI software
3. **Speech Generation**
    - Basic notions
    - Reproduced speech output
    - Time-dependent sound concatenation
    - Frequency-dependent sound concatenation
4. **Speech Transmission**
    - Signal form coding
    - Source coding in parameterized systems
    - Recognition and synthesis systems

Related Exam Questions:

- **2081 Exam**:
  How can you generate the speech in multimedia system? Explain.
- **2080 Exam**:
  How speech is generated? Describe.

Introduction to Sound

- **Sound** is a **mechanical wave** that travels through a medium (like air or water) and is **perceived by the human ear**.
- In multimedia, **sound is used** to enhance user interaction and provide a richer experience.
- It includes **speech, music, and sound effects**, used in games, animations, videos, and e-learning.
- For computers, **sound must be converted** from analog to digital to store and process it.

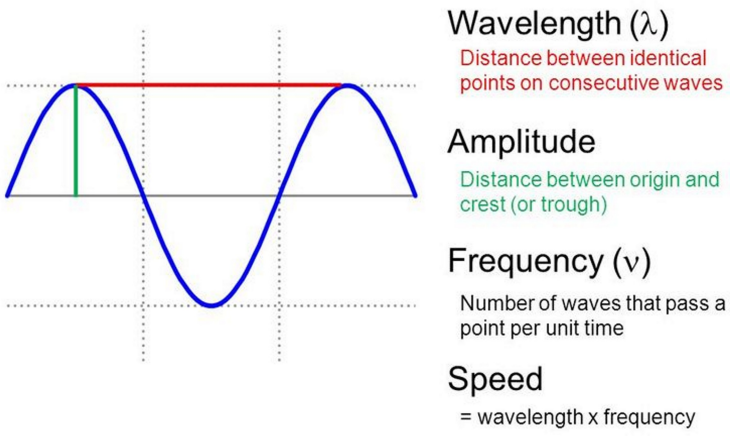
Basic Concepts of Sound

1. Frequency

- Number of sound wave cycles per second.
- Measured in **Hertz (Hz)**.
- Higher frequency = Higher pitch.
- Example: Human speech ≈ 300–3400 Hz.

2. Amplitude

- Height of the wave; indicates loudness.
- Measured in **decibels (dB)**.
- Higher amplitude = Louder sound.



Computer Representation of Sound

Concept:

- Sound in real life is **analog** — a continuous wave.
- Computers only understand **digital** data (0s and 1s).
- So, sound must be **converted** from analog to digital form using a process called **digitization**.

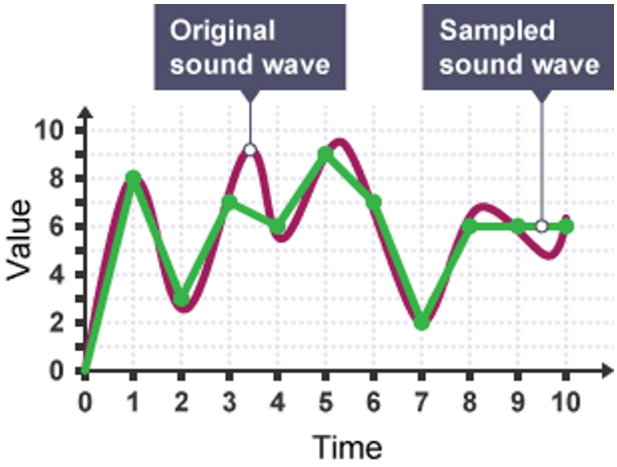
How it's done:

- The analog wave is measured at fixed intervals → **Sampling**
- Each sample's value is rounded off → **Quantization**
- Each value is stored in binary → **Encoding**

Example:

You speak into a microphone:

- Your voice is an analog wave.
- The mic converts it to an electrical signal.
- The computer **samples** it, **quantizes** the values, and stores it as **digital data** (e.g., in MP3 or WAV format).



Sampling Rate

Concept:

- It is **how many times per second** the sound is measured.
- Measured in **Hertz (Hz)**

Why it matters:

- Higher sampling rate = more accurate sound = larger file
- Lower sampling rate = poorer sound quality = smaller file

Rule:

- **Nyquist Theorem**: To record a sound accurately, the sampling rate should be **at least 2× the highest frequency** in the sound.

Example:

- Human ear hears up to **20,000 Hz**
- So, minimum sampling rate = **40,000 Hz**
- CD Audio uses **44,100 Hz (44.1 kHz)**

Quantization

Concept:

- After sampling, each sound sample has to be stored using a certain number of bits.
- Quantization means rounding the sample value to the nearest level that can be stored.

Bit Depth:

- 8-bit = 256 levels
- 16-bit = 65,536 levels
- More bits = better quality

Example:

Imagine you record a sound at 16-bit depth:

- Each sample is rounded to one of **65,536** possible values.
- This gives a very fine and accurate reproduction of the original sound wave.

Sound Hardware

Key components:

Hardware	Purpose	Example
Microphone	Captures analog sound	Voice input in phones, laptops
ADC (Analog-to-Digital Converter)	Converts analog input to digital	Inside sound cards
DAC (Digital-to-Analog	Converts digital back to analog	Sends sound to speakers

Converter)		
Speakers/Headphones	Output devices to hear sound	PC speakers, earbuds
Sound Card	Hardware that handles all sound processing	Integrated or external sound card

Basic MIDI Concepts

MIDI (Musical Instrument Digital Interface) is a communication protocol that allows electronic musical instruments, computers, and other related devices to connect and communicate with each other. It's not an audio format, but a way of transmitting data that represents musical information, such as pitch, velocity, and timing.

MIDI Devices

MIDI devices are the hardware or software components that send, receive, and/or process MIDI data. These devices allow electronic musical instruments, computers, and other musical devices to communicate with each other, facilitating music production and performance.

Types of MIDI Devices

1. **MIDI Controller**
    - A **MIDI controller** is a device that generates and sends MIDI messages, typically for controlling virtual instruments, sound modules, or digital audio workstations (DAWs).
    - **Examples**:
        - **MIDI Keyboards**: A piano-like keyboard that sends MIDI data when keys are pressed. It typically includes other controls like knobs, sliders, and pads for additional control over sound parameters (e.g., volume, pitch).
        - **Drum Pads**: A device with pads that can be hit like a drum kit, sending MIDI messages for each pad.

2. **MIDI Sound Module / Synthesizer**
    - A **MIDI sound module** or **synthesizer** receives MIDI data from a controller or computer

and generates sound based on the received instructions (e.g., note, pitch, velocity, modulation).
    - **Example**: A **hardware synthesizer** like the Roland Jupiter or Yamaha DX7 receives MIDI data and produces audio based on it.
3. **MIDI Interface**
    - A **MIDI interface** connects MIDI devices to a computer or other MIDI-enabled equipment. It allows MIDI data to be transmitted between the computer and external MIDI devices.
    - These interfaces can be **USB-to-MIDI** converters or **MIDI interfaces** with multiple input/output ports to connect many devices.
    - **Example**: The **MIDI USB interface** connects a keyboard to a computer so the DAW or software can record the MIDI performance.
4. **MIDI Sequencer**
    - A **MIDI sequencer** is a device or software that records, arranges, and plays back MIDI data in a structured timeline.
    - **Example**: Software sequencers like **FL Studio**, **Ableton Live**, or **Logic Pro** allow users to create and edit MIDI tracks, specifying which notes to play and when.

MIDI Messages

MIDI (Musical Instrument Digital Interface) doesn't carry actual audio. Instead, it sends **digital instructions** (called **MIDI messages**) that tell musical devices what to play, how to play, and when to play it.

- For example: "Play note C4 with piano sound at volume 80" — this is a MIDI message, not the sound itself.
- These messages are sent over **16 MIDI channels** and can control keyboards, synths, software, etc.

Types of MIDI Messages

MIDI messages are categorized into two main types:

1. Channel Messages

- These control the **musical performance** on a specific MIDI channel (1 to 16).
- Used to play notes, change instruments, adjust effects, etc.

Main Types & Examples:

Type	Purpose	Example
Note On	Start playing a note	"Play note C4 on channel 1 at velocity 100"
Note Off	Stop a note	"Stop note C4 on channel 1"
Program Change	Change instrument/patch	"Switch to Guitar on channel 3"
Control Change	Change settings (volume, pan)	"Set volume to 90 on channel 5"
Pitch Bend	Bend pitch slightly	"Bend note up on channel 2"

These are like telling a musician: "Play louder", "Switch to violin", etc.

2. System Messages

- These **control the whole MIDI system**, not tied to a single channel.
- Used for **timing, synchronization**, and special features.

Main Types & Examples:

Type	Purpose	Example

MIDI Start	Start playback	"Start the sequencer"
MIDI Stop	Stop playback	"Stop all music"
MIDI Clock	Synchronize timing	"Keep tempo synced between devices"
System Exclusive	Manufacturer-specific commands	"Send custom command to Yamaha keyboard"
MIDI Time Code (MTC)	Sync with video or film (SMPTE)	"Align music with video frames"

MIDI Software

MIDI software refers to programs that allow users to **create, edit, record, and play** MIDI data. These programs send MIDI messages to instruments or virtual devices, and receive MIDI data for recording or analysis.

Types of MIDI Software

Type	Description
DAWs (Digital Audio Workstations)	Used to compose, arrange, record, and edit MIDI and audio.
MIDI Sequencers	Specifically designed to record and play back MIDI events in sequence.
Virtual Instruments (VSTs)	Software-based instruments that respond to MIDI input to produce sound.
MIDI Editors	Allow you to view and edit MIDI notes/events manually (piano roll editors).

SMPTE Timing Standard

SMPTE (Society of Motion Picture and Television Engineers) timing standard is used to **synchronize audio (like MIDI) with video** in multimedia and film production.

Why SMPTE is needed

- MIDI is good for musical time (beats, bars, tempo).
- SMPTE is used for real-world **clock time** (hours, minutes, seconds, frames).
- SMPTE ensures **audio events match exact video frames**, making it ideal for scoring films, game sound design, or TV shows.

MIDI and SMPTE Synchronization

- MIDI Time Code (MTC) is a MIDI-compatible version of SMPTE.
- MTC allows a MIDI device to **stay in sync** with video playback.
- For example, you can set a MIDI sequence to start exactly at **00:01:30:00** in a film.

Example Use Case

A music producer scoring background music for a short film will use a **DAW** with **SMPTE timing** enabled, so that the drum hits or emotional piano lines **align perfectly with scene transitions** or actions in the video.

Speech Generation

Speech generation is the process of converting **text or data into spoken voice output** using computer systems.
This is a key part of **Text-to-Speech (TTS)** technology in multimedia systems.
It helps make systems more interactive and accessible — for example:

- Navigation apps reading directions aloud
- Screen readers for visually impaired users
- Virtual assistants like Siri or Google Assistant

Basic Notions of Speech Generation

These are the core concepts used when generating speech:

- **Phoneme:**
  The smallest unit of sound in a language.
  Example: the word “cat” has three phonemes: /k/, /æ/, /t/
- **Synthesis-by-rule:**
  Uses linguistic rules to convert text to phonemes, then to audio.
- **Text analysis:**
  Breaks input text into words, syllables, stress patterns, and punctuation to understand how it should sound when spoken.
- **Prosody (intonation):**
  Deals with the rhythm, stress, and pitch of speech. Helps make speech sound more **natural** and **human-like**.

Reproduced Speech Output

This refers to the **actual sound output** generated after processing text. There are two main methods:

Method	Description	Example
Pre-recorded Speech	Uses real human voice recordings stored as audio clips.	GPS system saying “Turn right” using pre-recorded audio.
Synthesized Speech	Generated by combining phonemes using a speech synthesizer.	Google Assistant reading a weather report in real-time.

Sound Concatenation in Speech Generation

Concatenation means *linking smaller sound segments together* to form complete speech.

1. Time-Dependent Sound Concatenation

Definition:

- This method links together *pre-recorded sound segments* based on their **timing and duration**.
- Each segment represents a small unit of speech like a phoneme, syllable, or word.
- The segments are selected and arranged **according to time** to form smooth, natural-sounding speech.

Example:

- The system stores audio clips of phonemes with different durations (like short or long "a").
- If a word like **"cat"** is spoken quickly, the system selects the shorter version of /æ/.
- If it's emphasized or spoken slowly, the longer /æ/ is chosen.
  Used in systems where **natural timing and rhythm** are important.

2. Frequency-Dependent Sound Concatenation

Definition:

- This method links sound segments based on their **pitch or frequency characteristics**.
- It adjusts the frequency of each segment to maintain a **consistent tone or intonation** in the generated speech.
- Useful when speech needs to match emotional tone or speaker identity.

Example:

- If the word "yes" is spoken in a high tone (e.g., for a question), the segments used are from high-frequency samples.
- For a statement, it might use a lower-pitched version.
  Used in systems that need to maintain **intonation and emotional expression**.

Comparison Table

Feature	Time-Dependent Concatenation	Frequency-Dependent Concatenation
Based on	Timing/duration	Pitch/frequency
Focus	Natural rhythm of speech	Tone, pitch, emotional expression
Used for	Fluent sentence construction	Expressive speech synthesis
Example	Short vs. long syllables	High-pitched vs. low-pitched voice

Speech Analysis

Introduction:

- Speech analysis refers to the *process of examining the properties of a speech signal* to extract meaningful features.

Purpose:

- To prepare the signal for further processing like recognition or compression.

Processes Involved:

- **Signal preprocessing**: removing noise.
- **Feature extraction**: pitch, energy, spectral components (e.g., using MFCC – Mel Frequency Cepstral Coefficients).
- **Segmentation**: dividing speech into phonemes or words.

Example:

- In voice assistants, analyzing the spoken input to break it down into sounds and extract key speech features.

Speech Recognition

Introduction:

- Speech recognition is the **conversion of spoken language into text** using software and algorithms.

Steps Involved:

1. **Input**: User speaks into a microphone.
2. **Preprocessing**: Signal is cleaned.
3. **Feature Extraction**: Convert speech into vectors using techniques like MFCC.
4. **Pattern Matching**: Compare against known speech patterns using models.
5. **Output**: Matched text is generated.

Technologies Used:

- **Hidden Markov Models (HMM)**
- **Deep Neural Networks (DNN)**

- **Natural Language Processing (NLP)**

Example:

- Saying “What’s the time?” to your phone and it converting it into text to respond.

Speech Transmission

Introduction:

- Speech transmission is the **sending of voice signals** from one point to another (e.g., phone calls, VoIP).

Techniques Used:

- **Signal form coding**: Preserves waveform shape (e.g., PCM).
- **Source coding**: Reduces data by removing redundancies (e.g., LPC – Linear Predictive Coding).

Types of Coding in Transmission:

Type	Purpose	Example
Signal Form Coding	Preserve waveform of the signal	PCM, ADPCM
Source Coding	Compress based on parameters	LPC, CELP

Example:

- During a Skype call, your voice is digitized, compressed, and transmitted over the internet.

Q. How is speech generated in a multimedia system?

(*Asked in 2080 & 2081 exam*)

Answer:

Speech generation is the process of **producing artificial speech output** using a computer. It involves generating human-like voice using various techniques. This is used in applications like virtual assistants, screen readers, and announcement systems.

Steps for Speech Generation in Multimedia System:

1. Basic Notions of Speech

- Speech is a **natural form of human communication**.
- It is made of **phonemes** (basic sound units), **intonation**, and **timing**.
- The computer needs to generate these elements to produce speech.

2. Reproduced Speech Output

- The computer **plays back pre-recorded human voice** (like audio clips).
- Suitable for **limited vocabulary systems** (e.g., ATM machines, IVR systems).
- Example: Saying "Welcome" or "Press 1 for English."

3. Time-dependent Sound Concatenation

- **Concatenation = joining sounds**.
- Here, small segments of recorded speech (like syllables or phonemes) are **joined over time**.
- The order and timing are managed to make smooth speech.
- Used in **Text-to-Speech (TTS)** systems.
- Example: TTS saying "Today is Monday" by joining sound pieces for "To", "day", "is", "Mon", "day".

4. Frequency-dependent Sound Concatenation

- Here, **pitch and frequency** of sounds are adjusted to match tone and expression.
- Makes speech sound more **natural and expressive**.

- Example: Adjusting the pitch to ask a question like "Are you okay?"

Diagram (optional in exams)

Text Input
↓
Text Analysis
↓
Phoneme Selection
↓
Sound Concatenation (Time/Frequency)
↓
Audio Output (via speakers)