Digital Audio

Lecture 05

BIL464 Multimedia Systems

Mustafa Sert
Asst. Prof.
msert@baskent.edu.tr

Department of Computer Engineering, Başkent University
Ankara 06810 TURKEY

Content

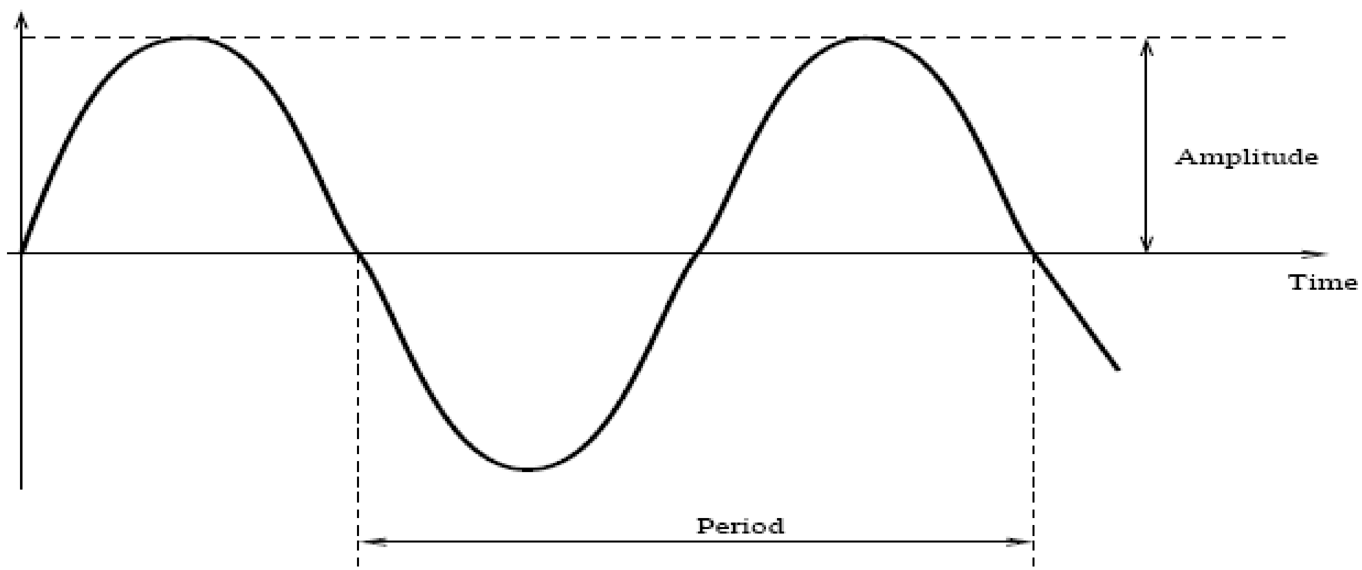
- The Nature of Sound/Audio
- Digital Sound Representation
  - Sampling Rate & Aliasing
  - Quantization
- Frequency Analysis
- Audio file formats
- Audio hardware

The Nature of Sound

- Sound is a physical phenomenon produced by the vibration of matter and transmitted as waves
- The term audio is synonymous with sound
- Sounds we heard everyday are very complex. Every sound is comprised of waves of many different frequencies and shapes. But the simplest sound we can hear is a sine wave:

The Nature of Sound – *Cont.*

- Sound waves can be characterized by the following attributes:
  - Period
  - Frequency
  - Amplitude
  - Bandwidth
  - Pitch
  - Loudness



Period, Pitch and Frequency

- Period *is the interval at which a periodic signal* repeats regularly
- Pitch *is a perception of sound by human beings It* measures how 'high' is the sound as it is perceived by a listener.
- Frequency *measures a physical property of a* wave. The unit is Herts (Hz) or kiloHertz (kHz).

Infra sound	0 – 20 Hz
Human hearing range	20 – 20 kHz
Ultrasound	20 kHz – 1 GHz
Hypersound	1 GHz – 10 THz

Loudness and Amplitude

- The other important perceptual quality is loudness or volume.
- Amplitude is the measure of sound levels. For a digital sound, amplitude is the sample value.
- The reason that sounds have different loudness is that they carry different amount of power

Bandwidth

- Bandwidth is the range of frequencies a device can produce or a human can hear.

FM radio	50Hz – 15kHz
AM radio	80Hz – 5kHz
CD player	20Hz – 20kHz
Sound Blaster 16 sound card	30Hz – 20kHz
Inexpensive microphone	80Hz – 12kHz
Telephone	300Hz – 3kHz
Children's ears	20Hz – 20kHz
Older ears	50Hz – 10kHz
Male voice	120Hz – 7kHz
Female voice	200Hz – 9kHz

Computer Representation of Sound

- Sound waves are continuous while computers are good at handling discrete numbers
- In order to store a sound wave in a computer, samples of the wave are taken
- Each sample is represented by a number, the ‘code’.
- This process is known as *digitization*
- This method of digitizing sound is know as *pulse code modulation (PCM)*.
- According to Nyquist sampling theorem, in order to capture all audible frequency components of a sound, i.e., up to 20 kHz, we need to set the sampling to at least twice of this. This is why one of the most popular sampling rate for high quality sound is 44100 Hz
- Another aspect we need to consider is the resolution, i.e., the number of bits used to represent a sample. Often, 16 bits are used for each sample in high quality sound

Digitization

- Digitization means conversion to a stream of numbers, and preferably these numbers should be integers for efficiency.

- Fig. 1 shows the 1-dimensional nature of sound: amplitude values depend on a 1D variable, time. (And note that images depend instead on a 2D set of variables, x and y).

Digitization – *Cont.*

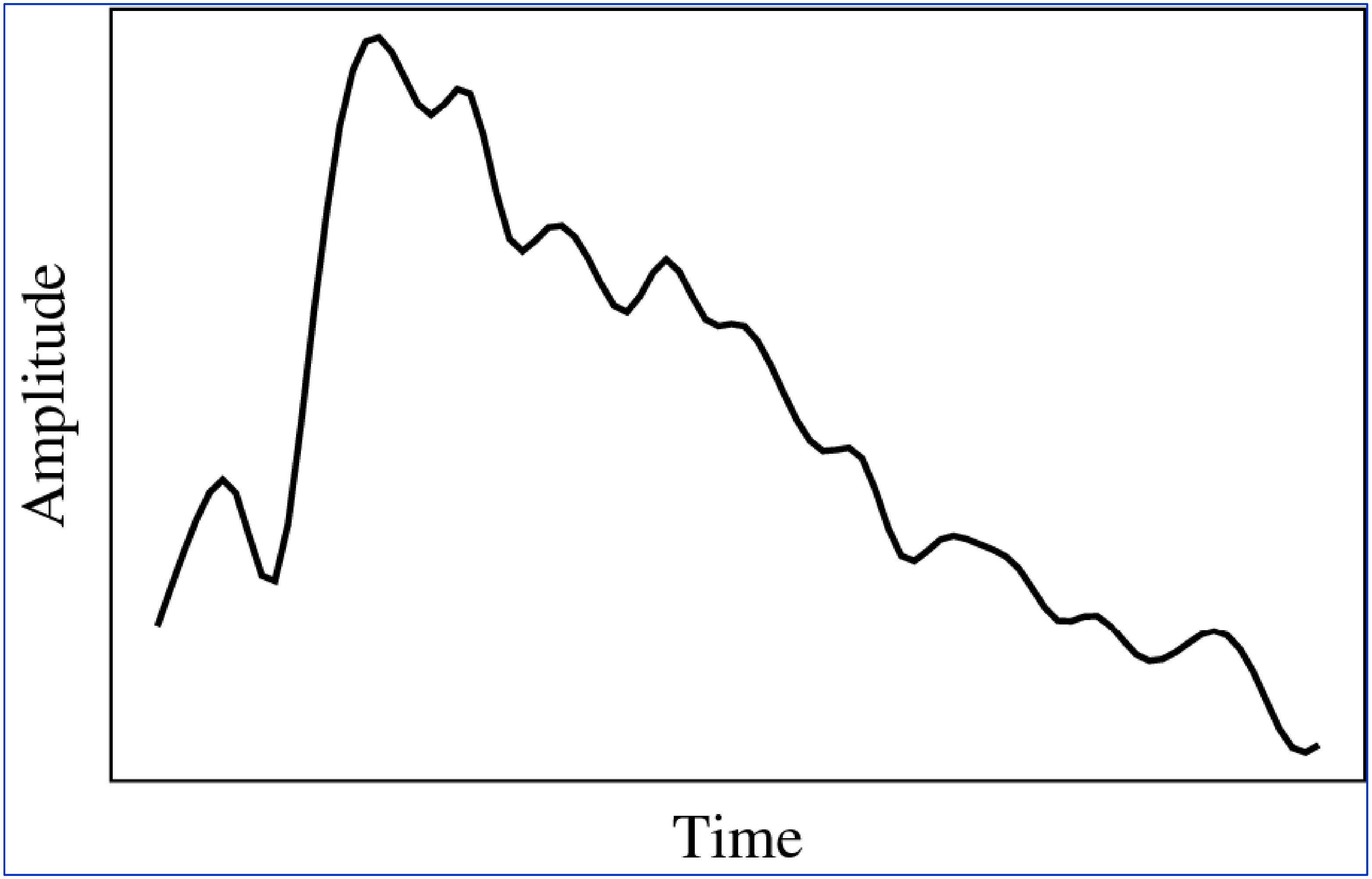
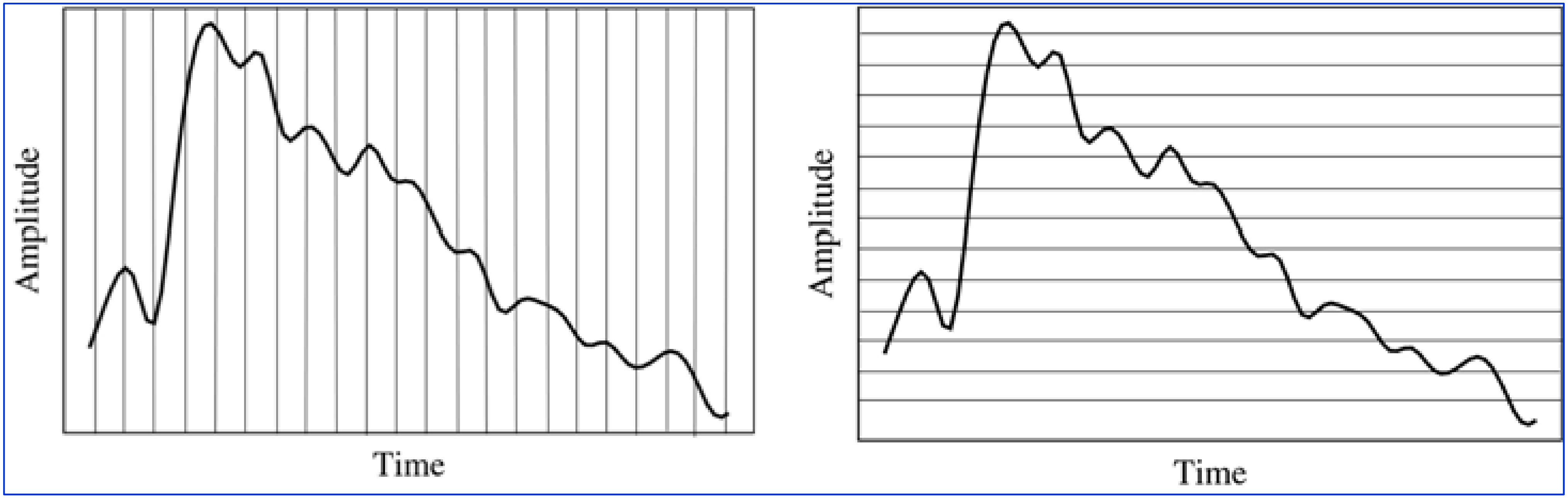


Fig. 1: An analog signal: continuous measurement of pressure wave.

Digitization – Cont.

- The graph in Fig. 1 has to be made digital in both time and amplitude. To digitize, the signal must be sampled in each dimension: in time, and in amplitude.
  - Sampling means measuring the quantity we are interested in, usually at evenly-spaced intervals.
  - The first kind of sampling, using measurements only at evenly spaced time intervals, is simply called, sampling. The rate at which it is performed is called the *sampling frequency* (see Fig. 2.a).
  - For audio, typical sampling rates are from 8 kHz (8,000 samples per second) to 48 kHz. This range is determined by the Nyquist theorem, discussed later.
  - Sampling in the amplitude or voltage dimension is called **quantization**. Fig.2.b shows this kind of sampling.

Digitization – Cont.



(a) (b)

Fig. 2: Sampling and Quantization. (a): Sampling the analog signal in the time dimension. (b): Quantization is sampling the analog signal in the amplitude dimension.

Digitization – *Cont.*

- Thus, to decide how to digitize audio data we need to answer the following questions:
  - What is the sampling rate?
  - How finely is the data to be quantized, and is quantization uniform?
  - How is audio data formatted? (file format)

Nyquist Theorem

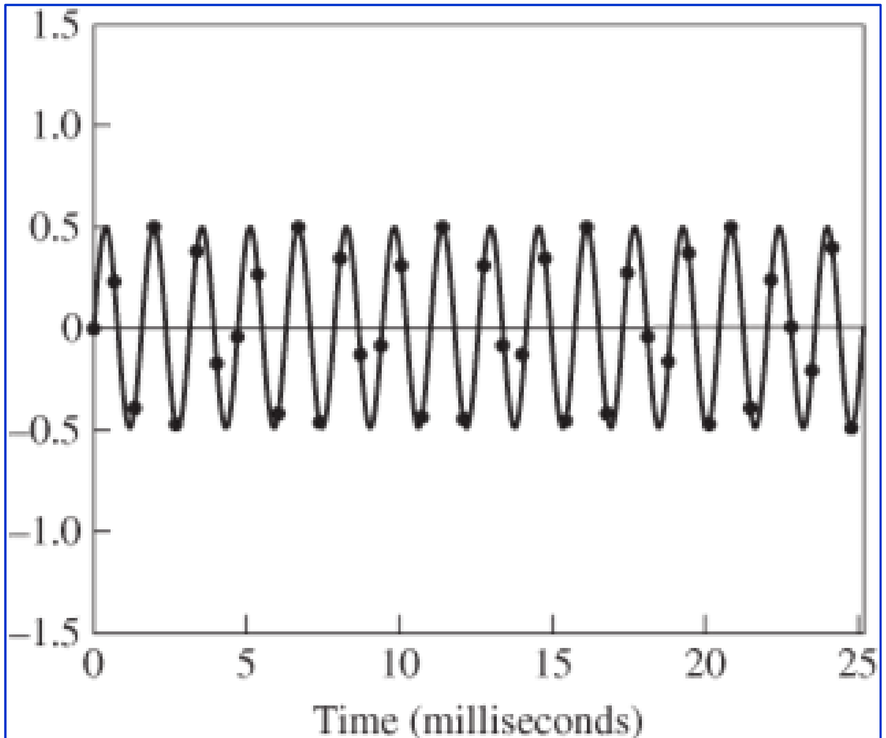
- The Nyquist theorem states how frequently we must sample in time to be able to recover the original sound.
  - Thus for correct sampling we must use a sampling rate equal to **at least twice the maximum frequency** content

KEY EQUATION

Given $f_{\max}$, the frequency of the highest-frequency component in an audio signal to be sampled, then the ***Nyquist rate***, f_{nr}, is defined as

$$f_{nr} = 2f_{\max}$$

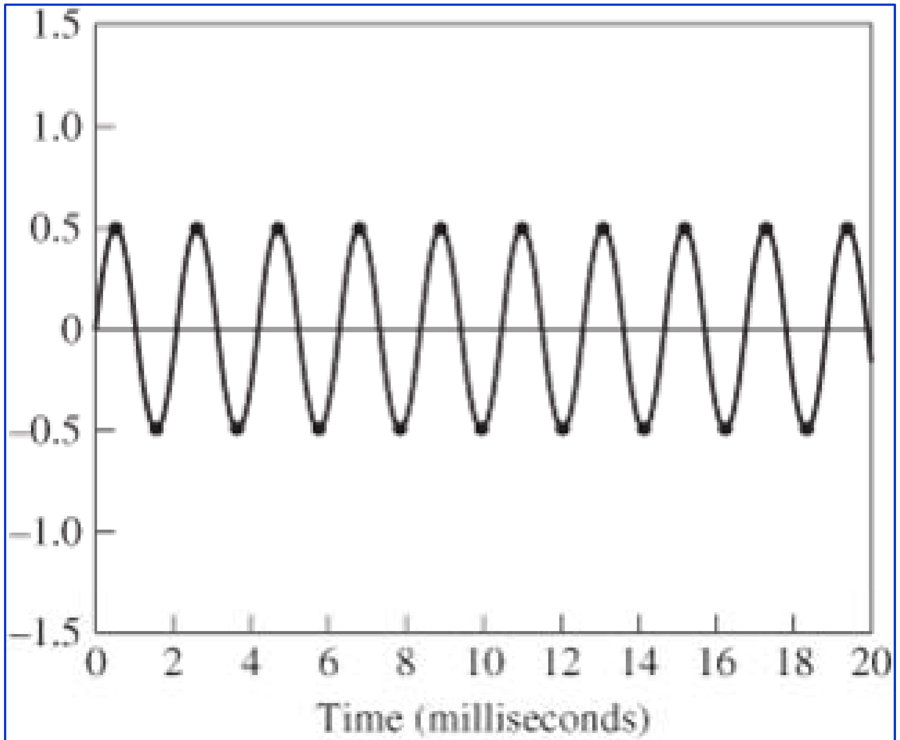
Nyquist Theorem – *Cont.*



(a)

- Samples taken more than twice per cycle will provide sufficient information to reproduced the wave with no aliasing
- If we take *more* than two samples per cycle on an analog wave, the wave can be precisely reconstructed from the samples, as shown in Figure

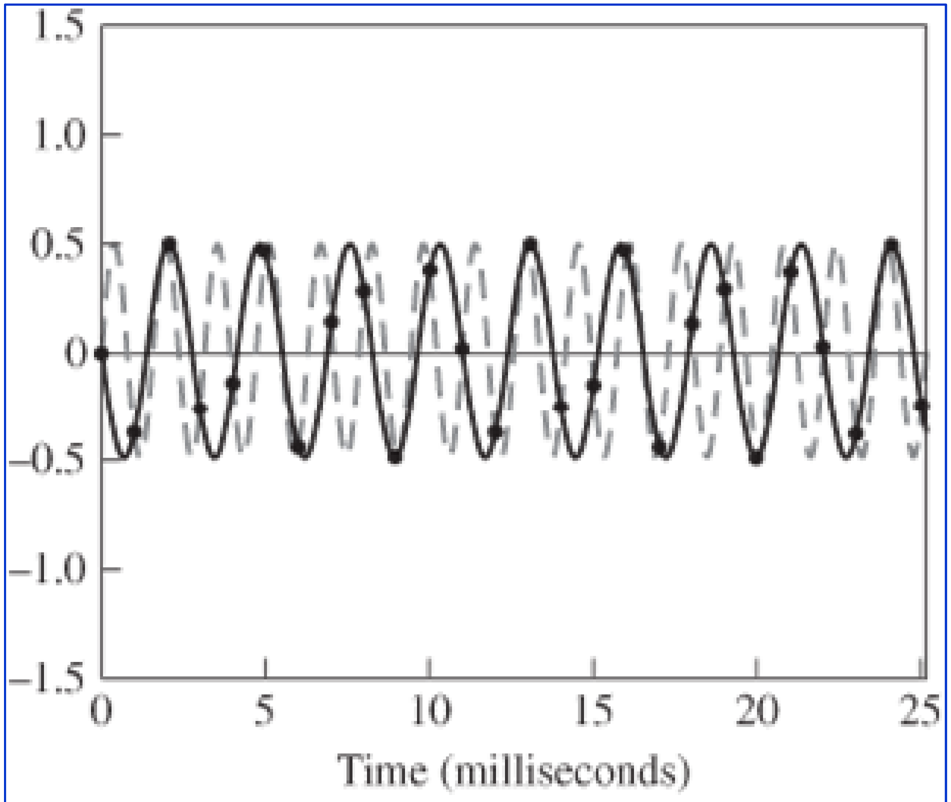
Nyquist Theorem – *Cont.*



(b)

- Samples taken exactly twice per cycle *can* be sufficient for digitizing the original with no aliasing
- If we have *exactly* two samples per cycle and the samples are taken at precisely the maximum and minimum values of the sine wave, once again the digitized wave can be reconstructed, as shown in the Figure. **<u>However</u>**, if the samples are taken at locations other than peaks and troughs, the frequency may be correct but the amplitude incorrect. In fact, the amplitude can be 0 if samples are always taken at 0 points.

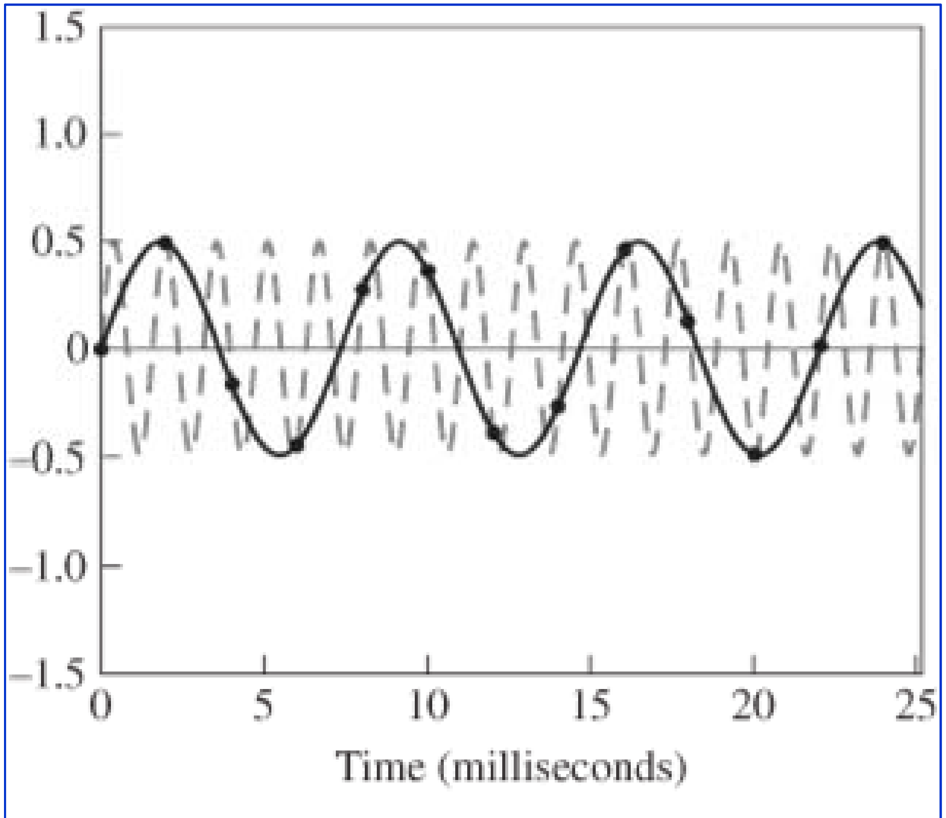
Nyquist Theorem – *Cont.*



(c)

- A wave sampled fewer than two times per cycle cannot be accurately reproduced. In the Figure, we see the result of sampling a 637 Hz wave at 1000 Hz, resulting in an aliased wave of 363 Hz. The inadequate sampling rate skips over some of the cycles, making it appear that the frequency of the actual wave is lower than it really is.
- The actual frequency is the sine wave drawn with the dashed line in the background.

Nyquist Theorem – *Cont.*



(d)

- This Figure shows a 637 Hz wave sampled at 500 Hz. Again, the sampling rate is below the Nyquist rate. In this case, the aliased wave has a frequency of 137 Hz.
- The actual frequency is the sine wave drawn with the dashed line in the background.

Alias Frequency

- The relationship among the Sampling Frequency, True Frequency, and the Alias Frequency is as follows:
  - **Case 1:** $f_{sampling} >= 2 \times f_{true}$
    - NO aliasing, original signal is recovered
  - **Case 2:** $f_{sampling}$ IN $[f_{true}, 2 \times f_{true})$
    - Aliasing occurs!
    - $f_{alias} = f_{sampling} - f_{true}$
  - **Case 3:** $f_{sampling} < f_{true}$
    - Aliasing occurs!

    (1) $f_{true}' = f_{true}$ MOD $f_{sampling}$

    (2) Follow the first two approaches for f_{true}'

Quality vs. File size

- The size of a digital recording depends on the sampling rate, resolution and number of channels:

  $$S = R * (b/8) * C * D$$

  S = file size (bytes), R = sampling rate (samples per second), b = resolution (bits), C = channels (1-mono, 2-stereo), D = recording duration (seconds)

- Higher sampling rate, higher resolution gives higher quality but bigger file size

Quality vs. File size – *Cont.*

- For example, if we record 10 seconds of stereo music at 44.1kHz, 16 bits, the size will be

  S = R * (b/8) *C * D

  S = 44100 * (16/8) * 2 * 10

  S = 1764000 bytes = 1722.7 Kbytes = 1.68 Mbytes

  Note:

  1 Kbytes = 1024 bytes

  1 Mbytes = 1024 Kbytes

- High quality sound files are very big, however, the file size can be reduced by compression

File Sizes for Common Sampling Rates

Sampling Rate	Resolution	Stereo /Mono	Size for for 1 Min.	Comments
44.1KHz	16-bit	Stereo	10.5MB	CD-quality recording
44.1KHz	16-bit	Mono	5.25MB	A good trade-off for high-quality recordings of mono sources such as voice-overs
44.1KHz	8-bit	Stereo	5.25MB	Achieves highest playback quality on low-end devices such as most of the sound cards
44.1KHz	8-bit	Mono	2.6MB	An appropriate trade-off for recording a mono source
22.05KHz	16-bit	Stereo	5.25MB	Darker sounding than CD-quality recording because of the lower sampling rate
22.05KHz	16-bit	Mono	2.5MB	Not a bad choice for speech, but better to trade some fidelity for a lot of disk space by dropping down to 8-bit
22.05KHz	8-bit	Stereo	2.6MB	A very popular choice for reasonable stereo recording where full bandwidth playback is not possible
22.05KHz	8-bit	Mono	1.3MB	A thinner sound than the choice just above, but very usable
11KHz	8-bit	Stereo	1.3MB	At this low a sampling rate, there are few advantages to using stereo
11KHz	8-bit	Mono	650K	In practice, probably as low as you can go and still get usable results
5.5KHz	8-bit	Stereo	650K	Stereo not effective
5.5KHz	8-bit	Mono	325K	About as good as a bad telephone connection

Frequency Analysis

- If you represent digital audio over the *time domain*, you store the wave as a one-dimensional array of amplitudes—the discrete samples taken over time.
  - This is probably the easiest way for you to think of audio data.
  - If you think of it as a function, the input is time and the output is a sample value.
- A complex waveform is in fact equal to an infinite sum of simple sinusoidal waves, beginning with a *fundamental frequency* and going through frequencies that are integer multiples of the fundamental frequency.
  - These integer multiples are called ***harmonic frequencies***.

Frequency Analysis – *Cont.*

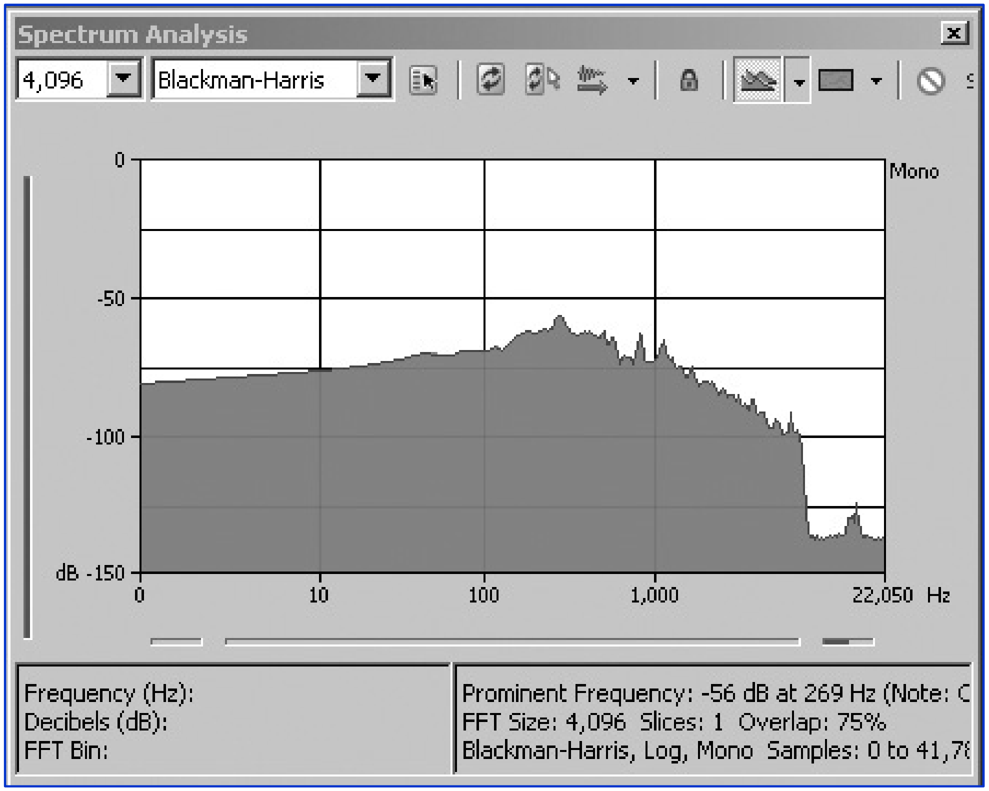
- To capture the complex waveform, it is sufficient to know the amplitude and phase of each of the component frequencies. The amplitude is, in a sense, how much each frequency component contributes to the total complex wave. This is how the data is stored in the *frequency domain*—as the amplitudes of frequency components.
  - If you think of this as a function, the input is frequency and the output is the magnitude of the frequency component.
  - It's useful to be able to separate the frequency components in order to analyze the nature of a sound wave and remove unwanted frequencies.
  - Notice that the time domain and frequency domain are equivalent. They both fully capture the waveform. They just store the information about the waveform differently.

Frequency Analysis – *Cont.*

- Audio processing programs provide information about the frequency components of an audio file. Two useful views of an audio file are the
  - **Frequency analysis view** and the **spectrogram view**.
- In the *frequency analysis view* (also called the *spectrum analysis*), frequency components are on the horizontal axis, and the amplitudes of the frequency components are on the vertical axis.
  - The frequency analysis view is useful for seeing, in a glance, how much of each frequency you have in a segment of your audio file.

Frequency Analysis – *Cont.*

- Notice that you can't take a frequency analysis at some instantaneous point in your audio file.
- Frequency implies a change in the amplitude of the wave over time, so some time must pass for there to be a frequency.



Frequency Analysis – *Cont.*

- The **spectrogram** is another alternative for showing frequency components. In a spectral view, time is on the horizontal axis and the frequency components are on the vertical axis, with the amplitude of the frequency components represented by color.
- Generally speaking, the brighter the color, the larger the amplitude for the frequency component. For example, blue could indicate the lowest amplitude for a frequency component. Increasingly high amplitudes could be represented by colors that move from blue to red to orange to yellow or white.

Frequency Analysis – *Cont.*

- The **spectrogram** view computes an "instantaneous spectrum" of frequencies for time *t* by applying the Fourier transform to a window on the audio data that surrounds *t*. The window is then moved forward in time, and the transform is applied again. Showing the frequency components in one view as they change over time can give you an easily understood profile of your audio data

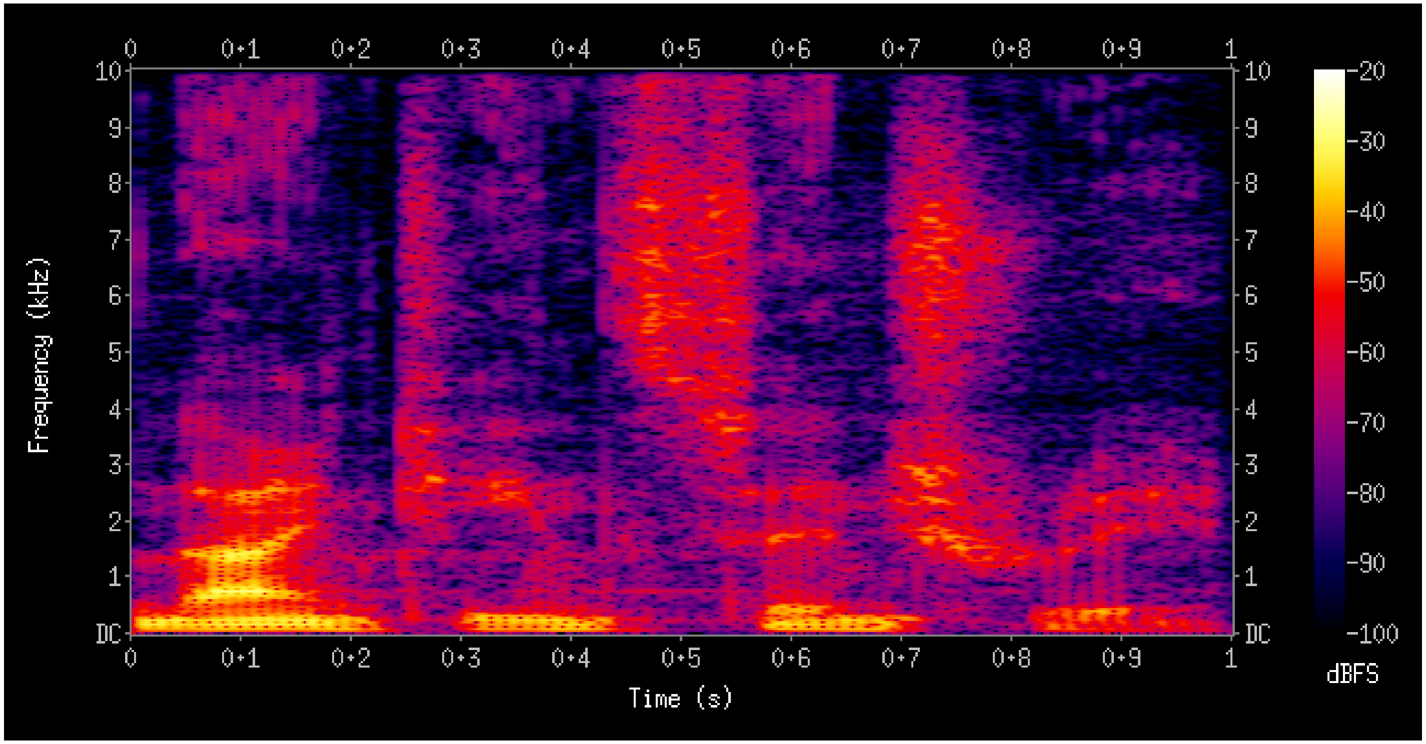


Fig. http://en.wikipedia.org/wiki/File:Spectrogram-19thC.png

Frequency Analysis – *Cont.*

- Any complex sinusoidal waveform is in fact a sum of simple sinusoidals can be written in the form of a Fourier series. A *Fourier series* is a representation of a periodic function as an infinite sum of sinusoidals:

$$f(t) = \sum_{n=-\infty}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)]$$

- The *discrete Fourier transform* (DFT) operates on an array of *N* audio samples, returning cosine and sine coefficients that represent the audio data in the frequency domain.
  - Let F_n be a discrete, complex number function representing a digital audio signal in the frequency domain. Let f_k be a discrete integer function representing a digitized audio signal in the time domain. $i = \sqrt{-1}$ and $\omega = 2\pi f$. Then the ***discrete Fourier transform*** is defined by

$$F_n = \frac{1}{N}\sum_{k=0}^{N-1} f_k \cos\left(\frac{2\pi nk}{N}\right) - i f_k \sin\left(\frac{2\pi nk}{N}\right) = \frac{1}{N}\sum_{k=0}^{N-1} f_k e^{\frac{-i2\pi nk}{N}}$$

$$\text{for } 0 \le n \le N-1$$

Audio File Formats

- The most commonly used digital sound format in Windows systems is .wav files
- Sound is stored in .wav as digital samples known as *Pulse Code Modulation* (PCM).
- Each .wav file has a header containing information of the file
  - Type of format, e.g., PCM or other modulations
  - Size of the data
  - Number of channels
  - Samples per second
  - Bytes per sample
- There is usually no compression in .wav files

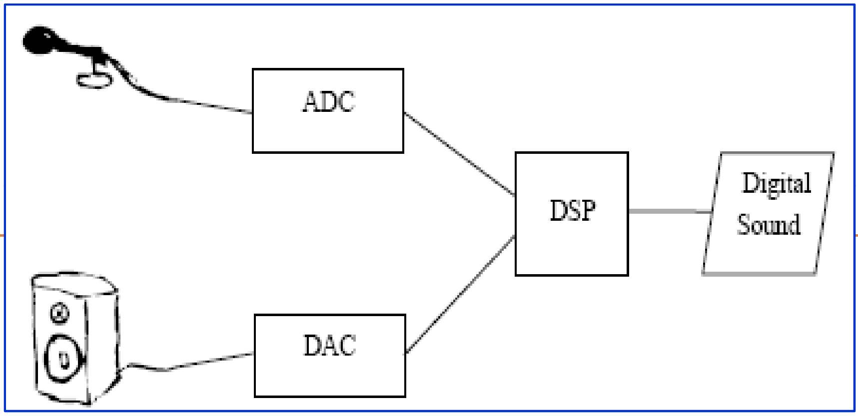
Audio File Formats – *Cont.*

- Other format may use different compression technique to reduce file size
- .vox use *Adaptive Delta Pulse Code Modulation (ADPCM)*.
- .mp3 MPEG-1 layer 3 audio.
- RealAudio file is a proprietary format

Audio File Formats – *Cont.*

Extension	MIME Type	Platform	Use
aif	Audio/x-aiff	Mac, SGI	Audio
aifc	Audio/x-aiff	Mac, SGI	Audio (compressed)
AIFF	Audio/x-aiff	Mac, SGI	Audio
aiff	Audio/x-aiff	Mac, SGI	Audio
au	Audio/basic	Sun, NeXT	ULAW audio data
mov	Video/QuickTime	Mac, Win	QuickTime video
mpe	Video/mpeg	All	MPEG video
mpeg	Video/mpeg	All	MPEG video
mpg	Video/mpeg	All	MPEG video
mp3	Audio/x-mpeg	All	MPEG audio
qt	Video/QuickTime	Mac, Win	QuickTime video
ra,ram	Audio/x-pn-realaudio	All	RealAudio Sound
snd	Audio/basic	Sun, NeXT	ULAW Audio Data
vox	Audio/	All	VoxWare Voice
wav	Audio/x-wav	Win	WAV Audio

Audio Hardware



- Recording and Digitising Sound
  - An *analog-to-digital converter (ADC)* converts the analog sound signal into digital samples.
  - A *digital signal processor (DSP)* processes the sample, e.g. filtering, modulation, compression, and so on.
- Play back Sound
  - A *digital signal processor processes the* sample, e.g. Decompression, demodulation, and so on
  - A Digital-to-Analog Convertor (DAC) converts the digital samples into sound signal
- All these hardware devices are integrated into a few chips on a sound card