11-755— Spring 2021
Large Scale Multimedia Processing

Lecture 1/6

Multimedia capture and storage

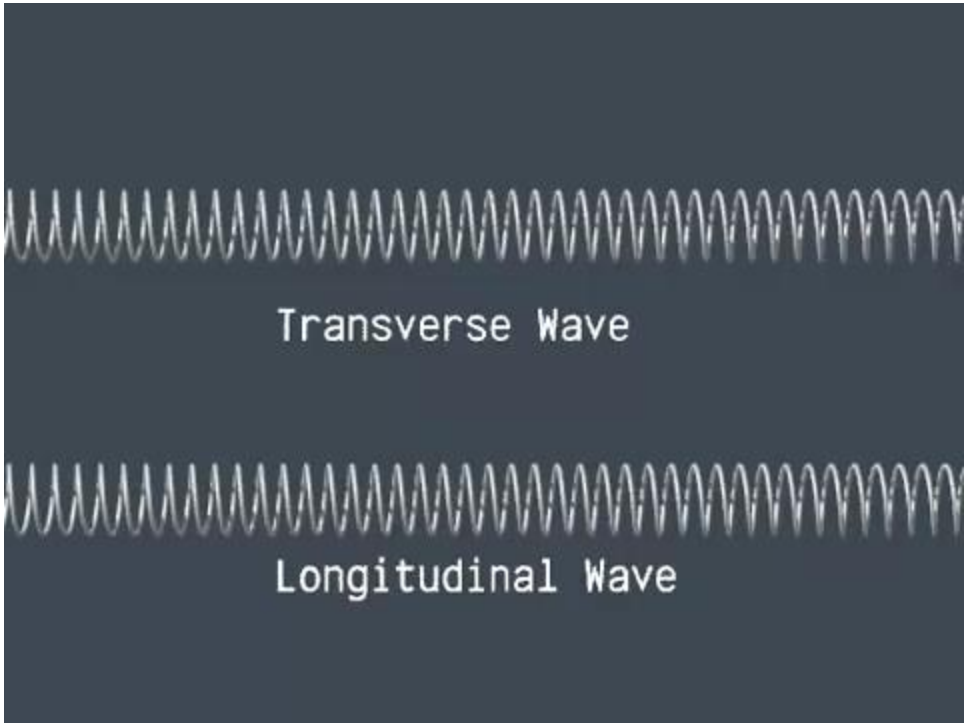
Rita Singh

Carnegie Mellon University

In this lecture

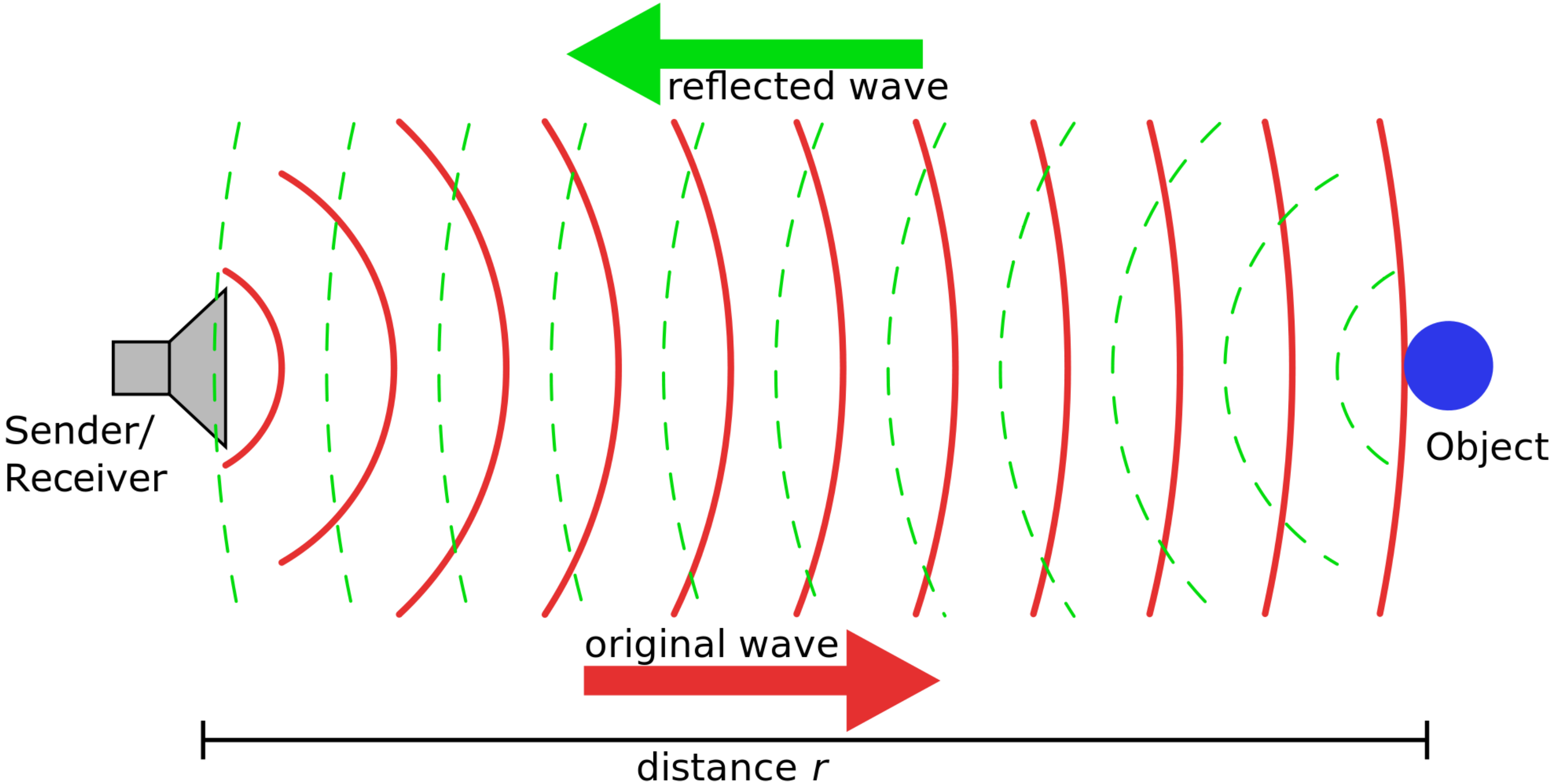
- Digital multimedia: Recording and devices
  - Audio
  - Images
  - Video
  - Text
- Digital multimedia: Processing
  - Audio processing
  - Two generic processing techniques

Sound waves



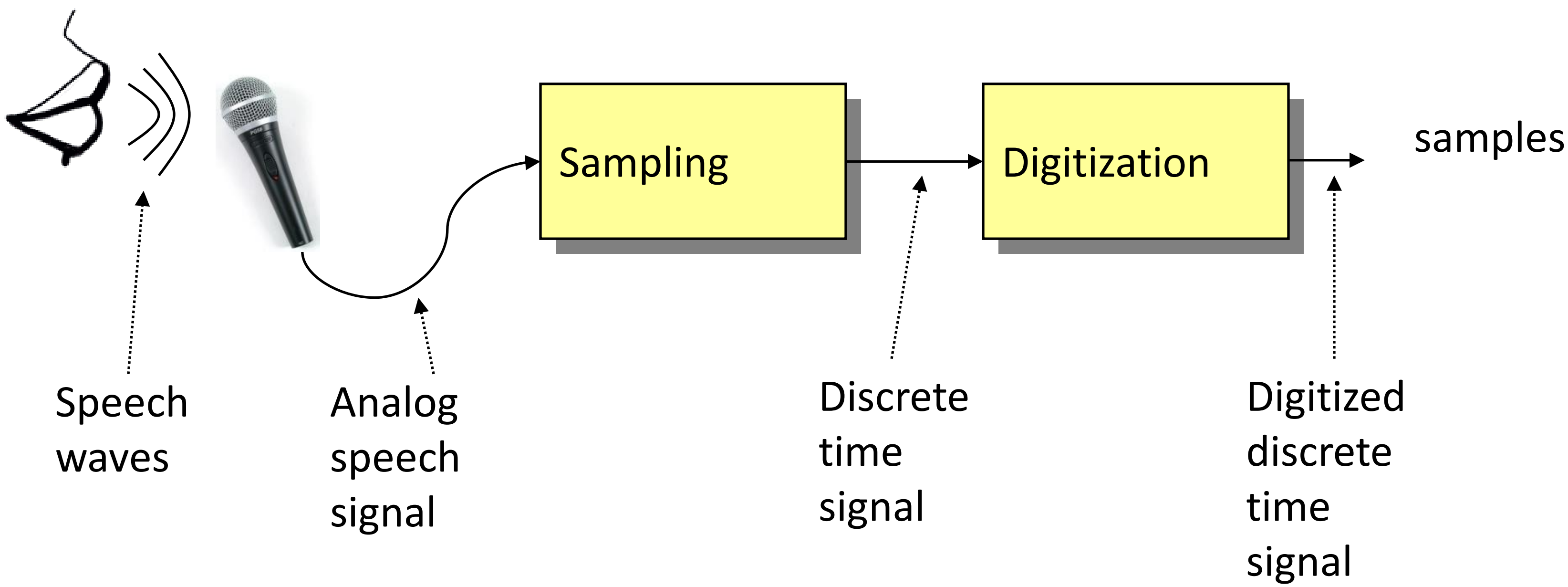
Sound is a longitudinal wave

What is sound?



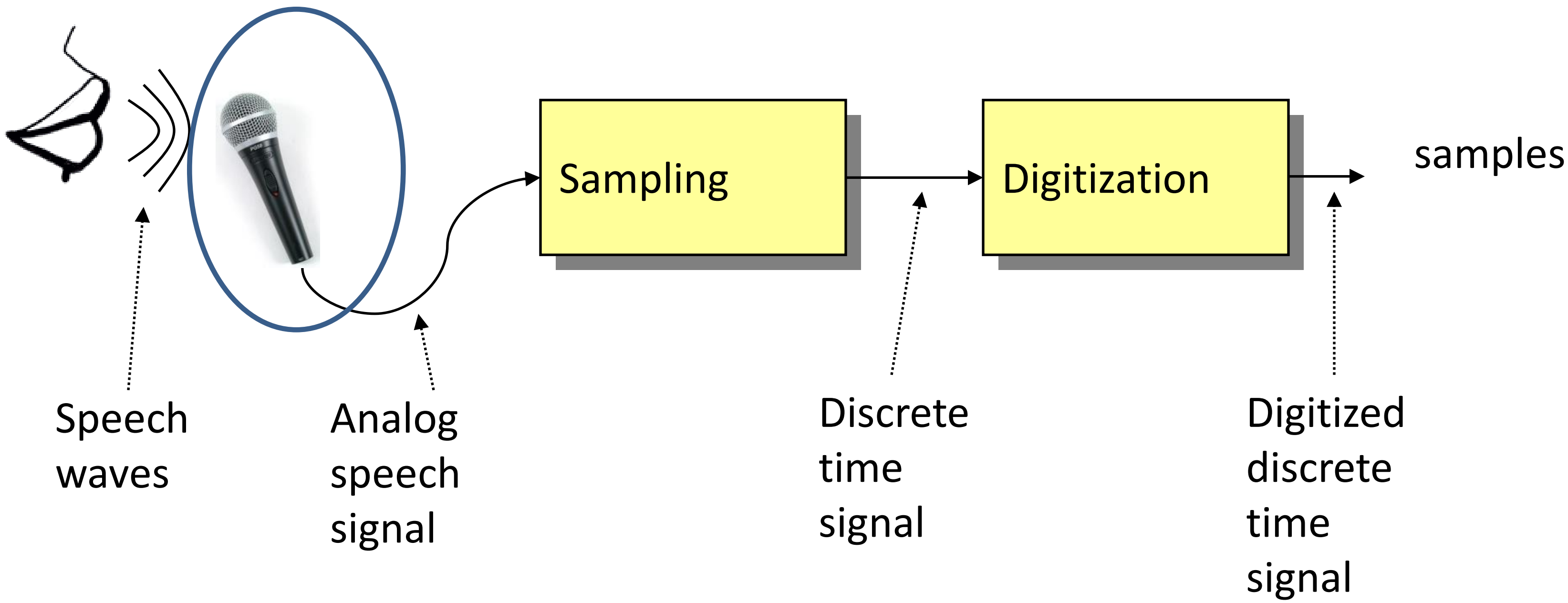
- All sounds are actually pressure waves
  - Sound exerts physical force

Audio Signal Capture



- The goal of signal capture is to convert pressure waves to a series of numbers that the computer can process
- Three stages
  - Transduction by a physical device
  - Sampling
  - Digitization

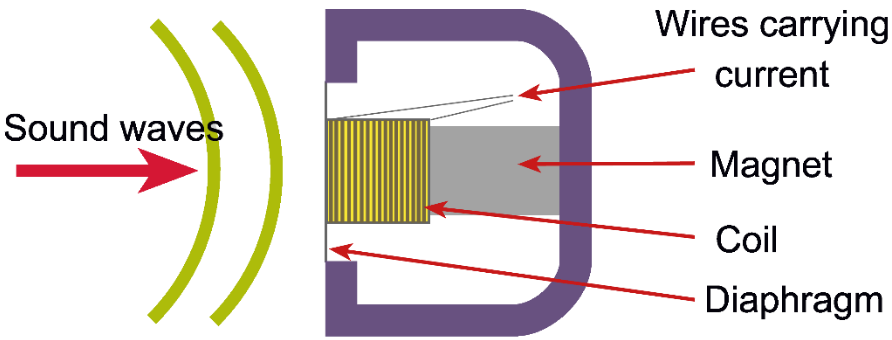
First Stage: Microphones

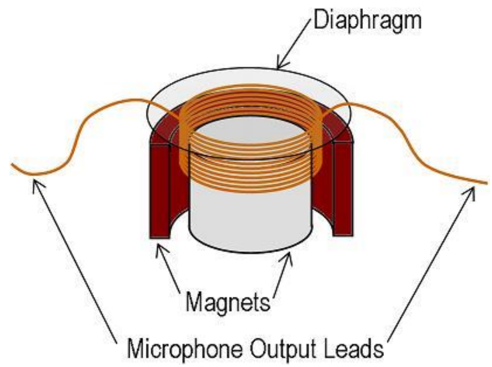


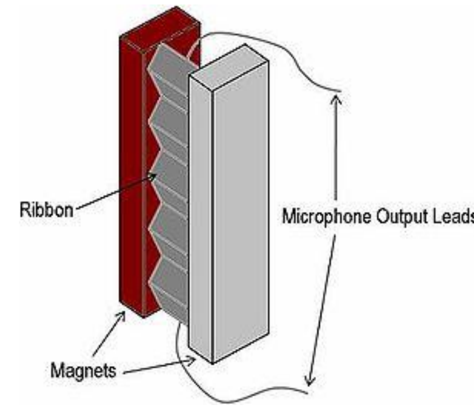
Microphones:

– A membrane is moved by the pressure wave
– A transduction mechanism that converts the movement of the membrane to an electrical signal
  • Based on inductance, capacitance or resistance
  • The signal is then sampled and digitized

Dynamic Mics: Inductive







- Based on inductance
- Time-varying voltage levels created by moving an element in an electromagnetic field
  - Moving coil mic: Diaphragm connected to a coil, suspended between magnets
    - Movement of diaphragms moves the coil and generates a voltage
  - Ribbon mic: A metal ribbon suspended between magnets

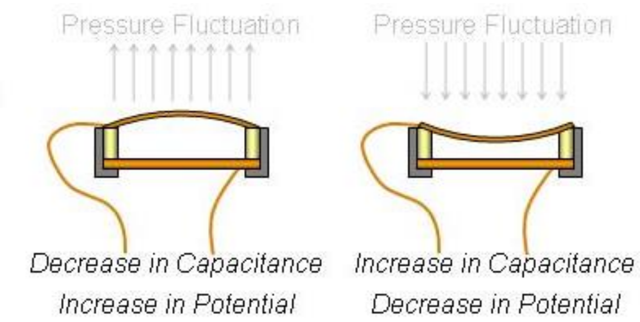
- Does not require phantom power
- Frequency response generally not flat
  - Also, small mics may be noise prone

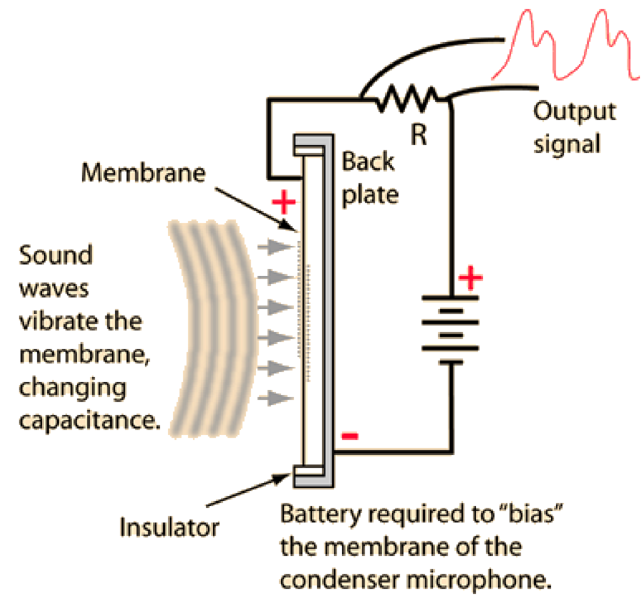
The First Microphone

- Graham Bell's "liquid" microphone
  - Metal cup filled with water, mixed with Sulphuric acid
  - Needle suspended in water (stuck to a diaphragm)
  - Sound waves move the needle up and down
    - Changing the resistance of the cup
  - A fixed voltage induces a time-varying current

- The design was actually "copied" from Elisha Gray
  - Apparently from a drawing Bell saw in the patent office
  - In later incarnations he discarded the liquid microphone

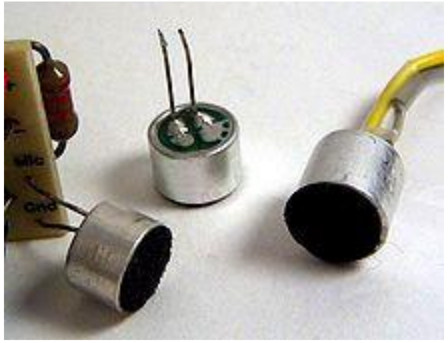
Condenser Microphones: Capacitance based

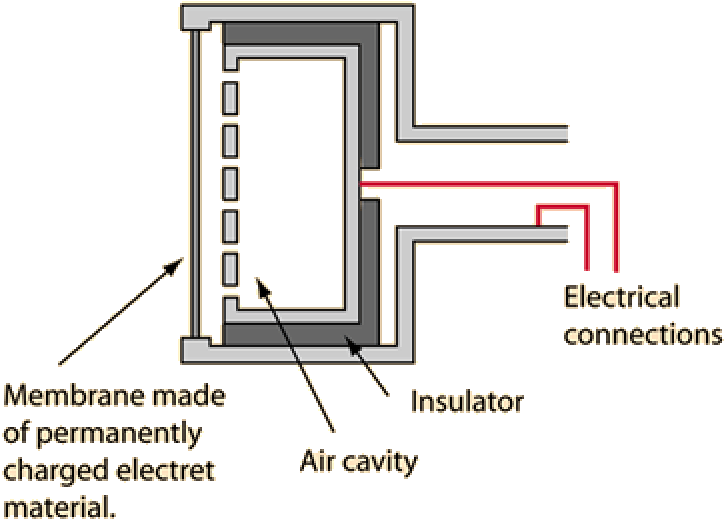




- Diaphragm and back plate maintained at different charge levels
- Pressure waves change the thickness of intervening dielectric (air)
  - Changing the capacitance of the capsule
  - Thereby changing the potential difference across the plates
- The changing potential difference is measured across the large resistor

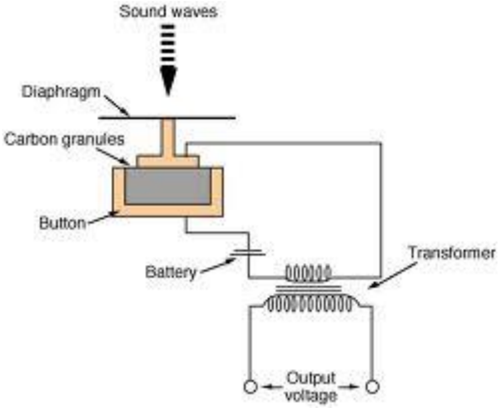
Electret Microphones: Capacitance based





- Electret microphones are Condenser mics with fixed charge level
  - Either the diaphragm or the backplate carry a fixed charge
  - No external voltage source needed to maintain charge
    - But do require power for preamp (usually DC source)
  - Used to be low quality, no longer
- Most computer microphones are electrets

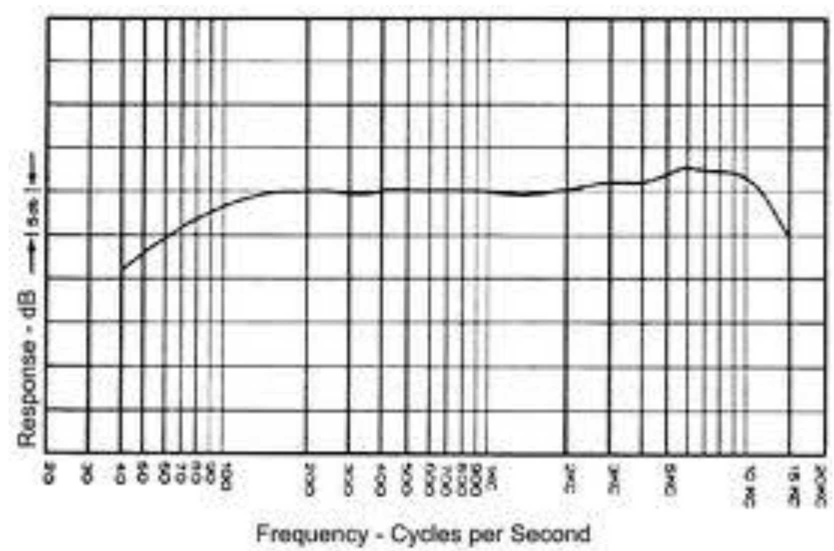
Carbon Button Microphones: Resistance based



- Based on *resistance*
- Carbon granules between a diaphragm and a backplate
- Motion of the diaphragm changes the resistance of the granule layer
- This changes the current in the circuit
  - Typically transformed to a higher voltage as shown

- Typical in older telephone handsets
  - Cellphones and recent handsets use electrets

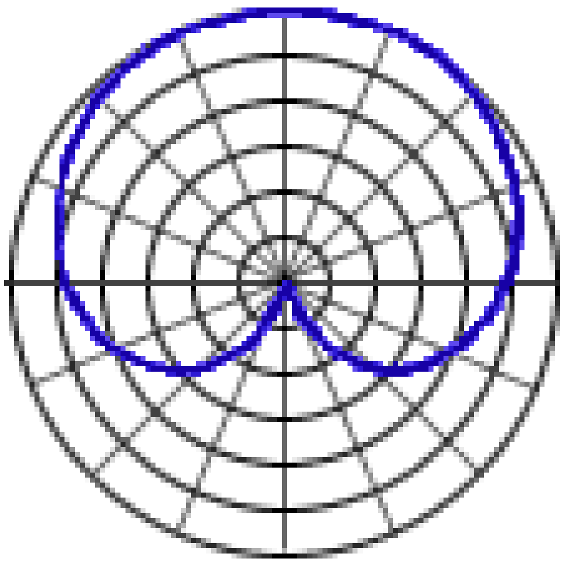
Choice of Microphones



- Characteristics to look for:
  - Frequency response
    - Must be flat in the frequencies of interest
  - Directivity
    - More on this later
  - Noise level
    - Usually stated as dB(A) SPL (Indicates the noise floor of the mic)
    - Lower is better
      - Good microphones: 20dB SPL, Ultra very good mics: 0dB
- Frequency response: Condenser microphones are usually used for high-quality recordings
  - Even cheap electret microphones can be surprisingly good

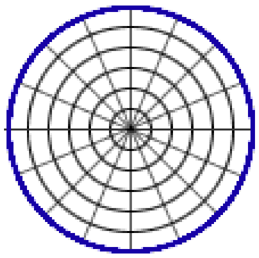
The "Directionality" of a Microphone

- The response of a microphone depends on the direction of the sound source
- This depends on the construction of the microphone, as well as the shape of its chassis
- The response is typically represented as a polar plot
  - The distance of the curve from the "center" of the plot in any direction shows the "gain" of the mic to sounds from that direction

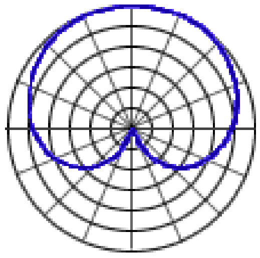


Typical Directivity Patterns

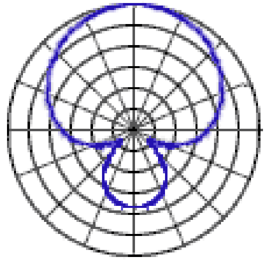
- Omnidirectional
  - Picks up sound uniformly from all directions



- Cardioid
  - Picks up sound from a relatively wide angle of directions from the front and some sound from the side



- Hyper cardioid
  - Picks up sound from a relatively narrow angle to the front, but also some noise from behind



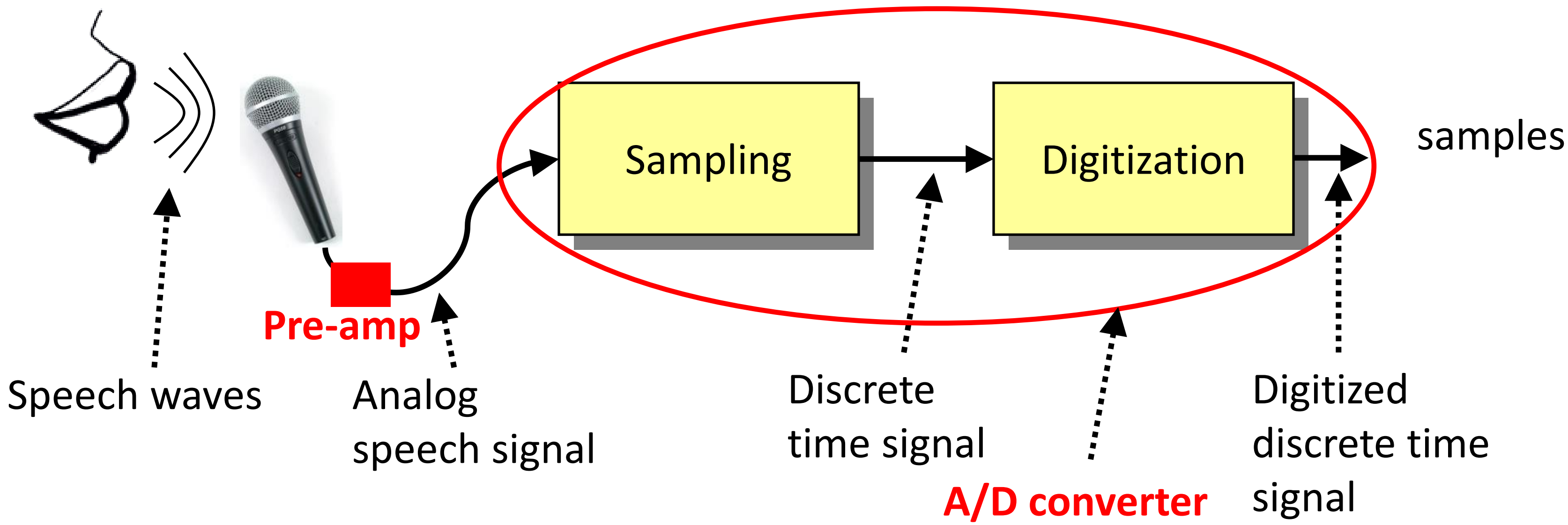
Directional Patterns

- Omnidirectional microphones are useful when we wish to pick up sounds from all directions
  - For speech applications, however, this will result in picking up a lot of noise from non-speech directions
- Where the speaker location can be somewhat localized, Cardioid microphones are more precise
  - Pick up less non-speech noise
  - Still susceptible to noise
- Hypercardioids are a better choice when the speaker location can be well localized
  - Risk of picking up noise from behind
- In general, utilizing microphone directionality is the best way of minimizing extraneous noise from recordings

Wearing/positioning the microphone

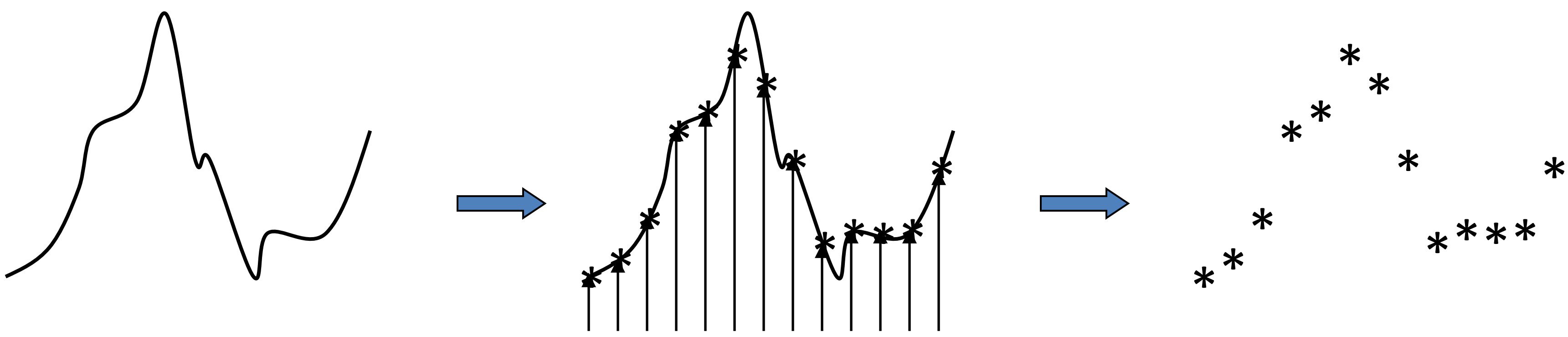
- Microphone must ideally be close enough for high speech level
  - Speech level significantly greater than background noise
- Positioning it too close can result in puffs of air from mouth saturating the mic
  - E.g. sounds such as “P”, “F”, etc.
- To avoid this, desktop mics frequently use a wind sock
  - A foam/sponge covering
- Head-mounted mics must be placed slightly to the side
  - Away from the direct path of puffs

Sampling and Digitization



- The signal from the microphone goes through a pre-amp
  - The gain of which can be adjusted
  - The output of the pre-amp is a continuous-time electrical signal
    - Usually a voltage signal
- The signal is digitized by an analog to digital converter
  1. **SAMPLING**: The signal value is sensed at regular, periodic intervals of time
  2. **DIGITIZATION/QUANTIZATION**: The value at each instant is assigned to one of a number of fixed values

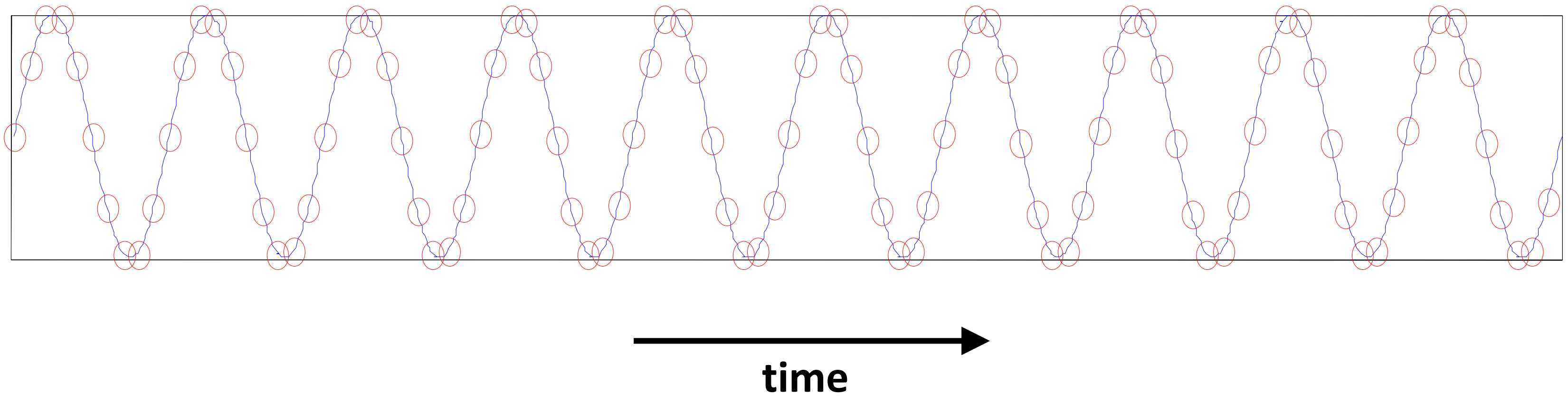
Sampling



- Sampling
  - Signal values are captured at discrete intervals of time
- Sampling requirement
  - The sequence of numbers must be sufficient to reconstruct the original signal perfectly, i.e. must retain all the information in the signal
    - When the analog signal is recreated from the digital signal, it should be exactly like the original signal
      - Sample values are converted to levels of an electrical signal
      - The electrical signal moves a diaphragm back and forth to produce a pressure wave, sensed as sound
      - Played out on a speaker

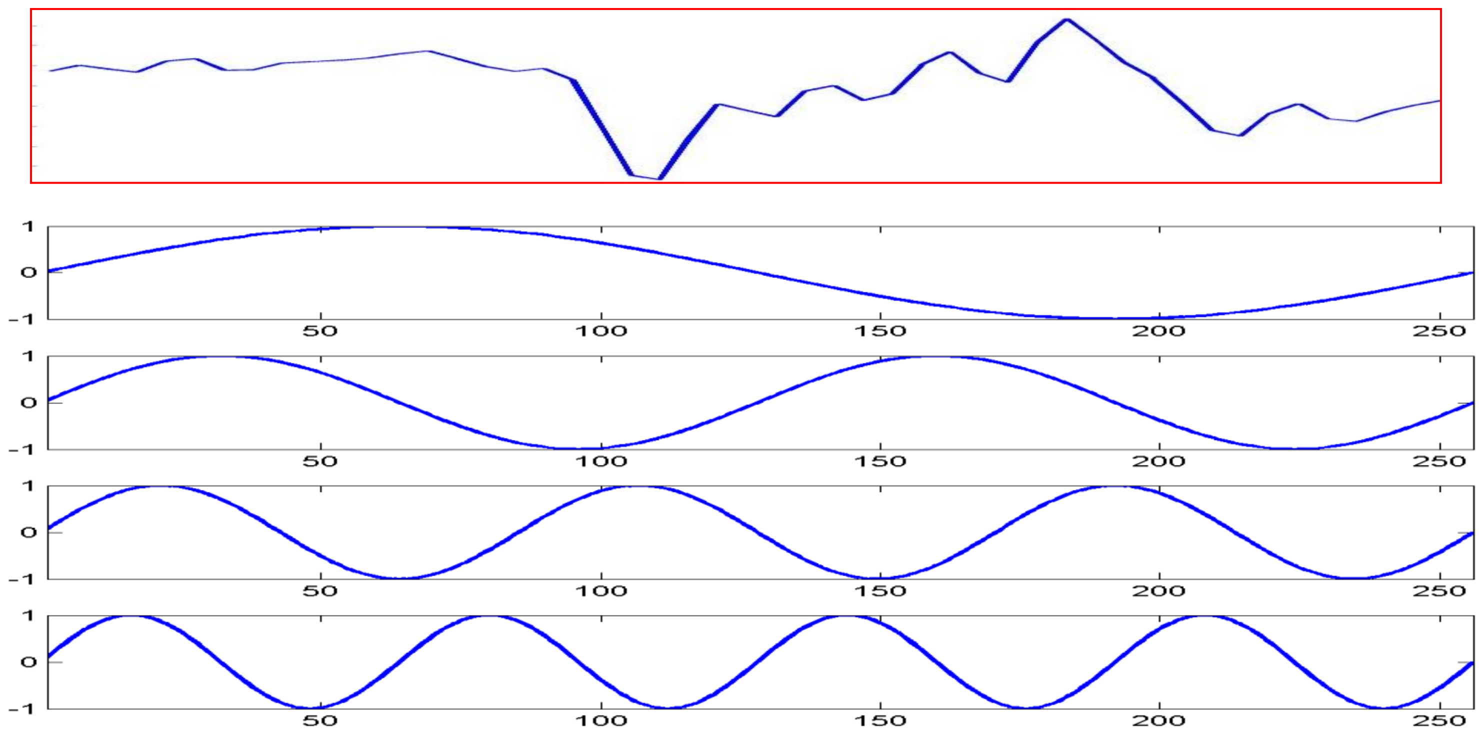
Sampling frequency

- *Sampling frequency (or sampling **rate**)* is the number of samples taken per second
  - Usually measured in hertz (Hz) : 1 Hz = 1 sample per second
- **Sampling rate should be such that all frequencies in the analog signal are accurately represented in the digital signal**



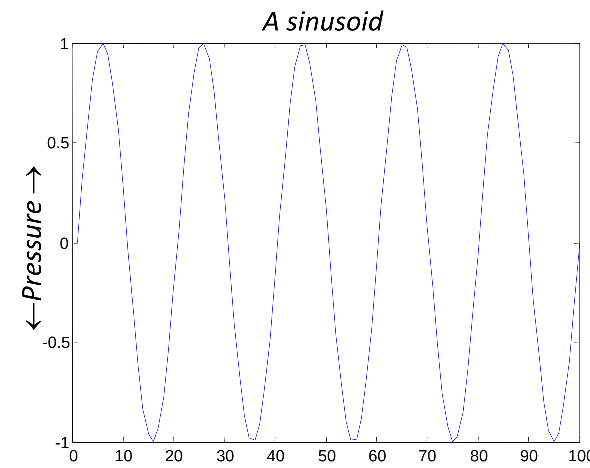
Sampling frequency

- Fourier Theorem: Any signal can be represented as a sum of sinusoids

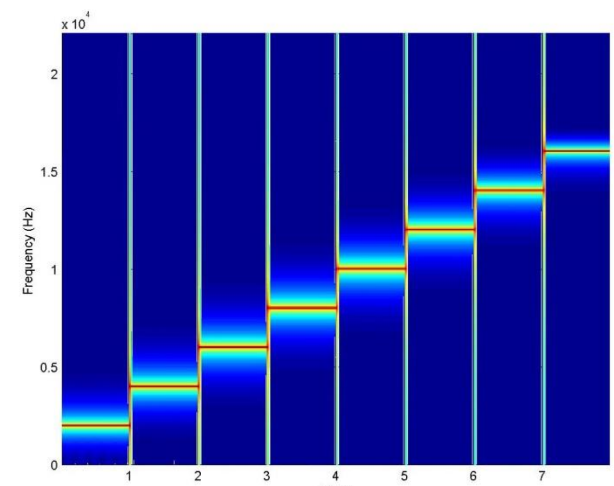


How many samples a second

- Sounds can be modeled as the sum of many sinusoids of different frequencies
  – Each hair cell in our inner ear is tuned to specific frequency



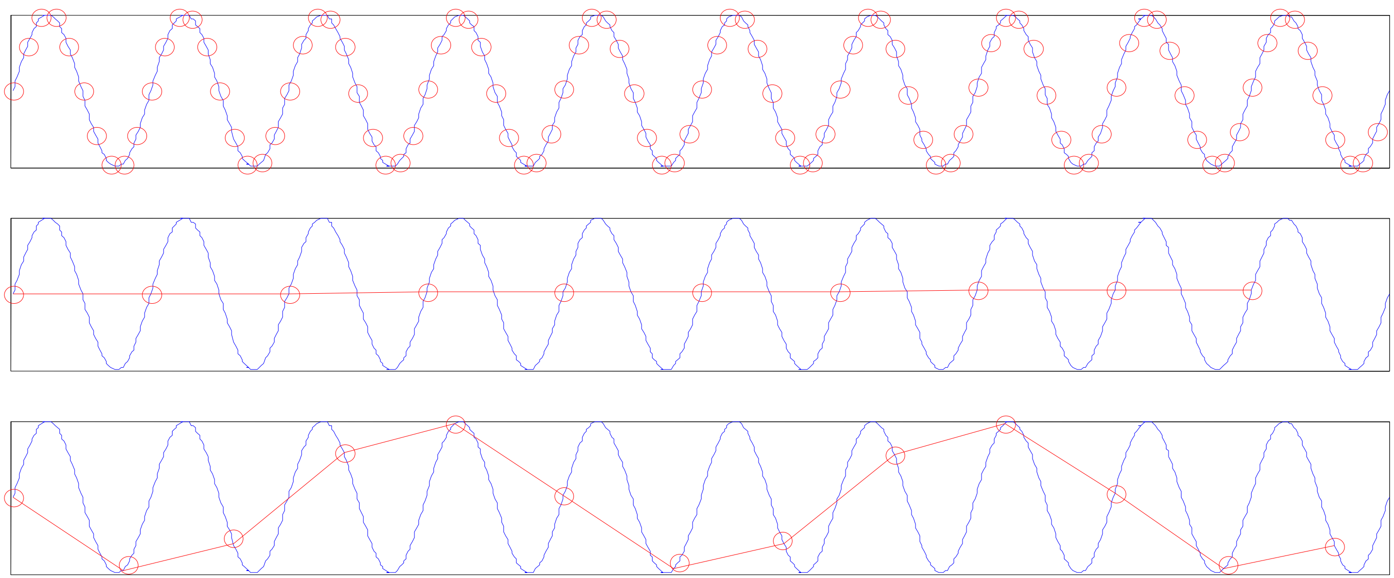
- Human hearing range
  – We can hear frequencies up to 16000Hz
  – Frequency components above 16000Hz can be heard by children and some young adults
  – Nearly nobody can hear over 20000Hz.



Aliasing

- **Sampling rate should be such that all frequencies in the analog signal are accurately represented in the digital signal**
- Low sampling rates result in *aliasing*
  - High frequencies appear as low frequencies
- **NYQUIST THEOREM:** To represent any frequency *F* accurately, we must sample at the rate of at least 2*F* samples per second

Aliasing: Visual Examples Signal is samples at less than 2F in each case



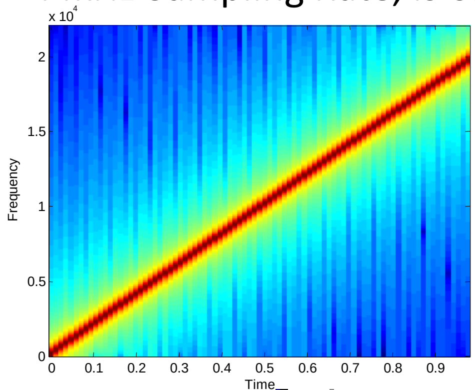
Aliasing

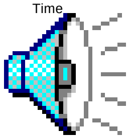
- The **Nyquist rate** or frequency is the minimum **rate** at which a finite bandwidth signal must be sampled to accurately retain all represent all component frequencies
  - For a signal with max frequency F, **Nyquist** frequency N is 2F.
- Low sampling rates result in *aliasing*
  - Frequency ((sampling rate)/2 + f) will appear as ((sampling-rate)/2 $-$ f)
    - e.g. 8000 sampling, 5k Hz will appear as 3 k Hz
    - For best accuracy, sampling rate is ideally N

Aliasing examples

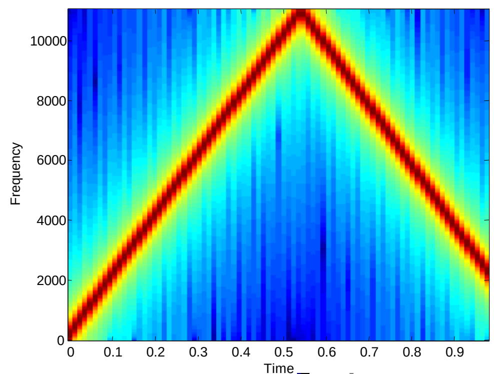
Sinusoid sweeping from 0Hz to 20kHz (a Chirp signal)

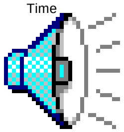
44kHz Sampling Rate, is ok



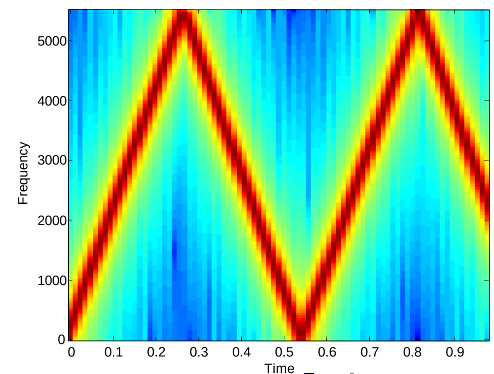


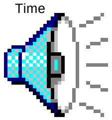
22kHz SR, aliasing!





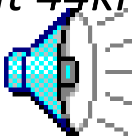
11kHz SR, double aliasing!



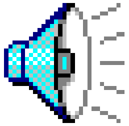


On real sounds

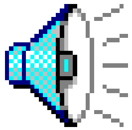
at 44kHz



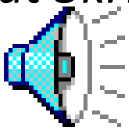
at 22kHz



at 11kHz



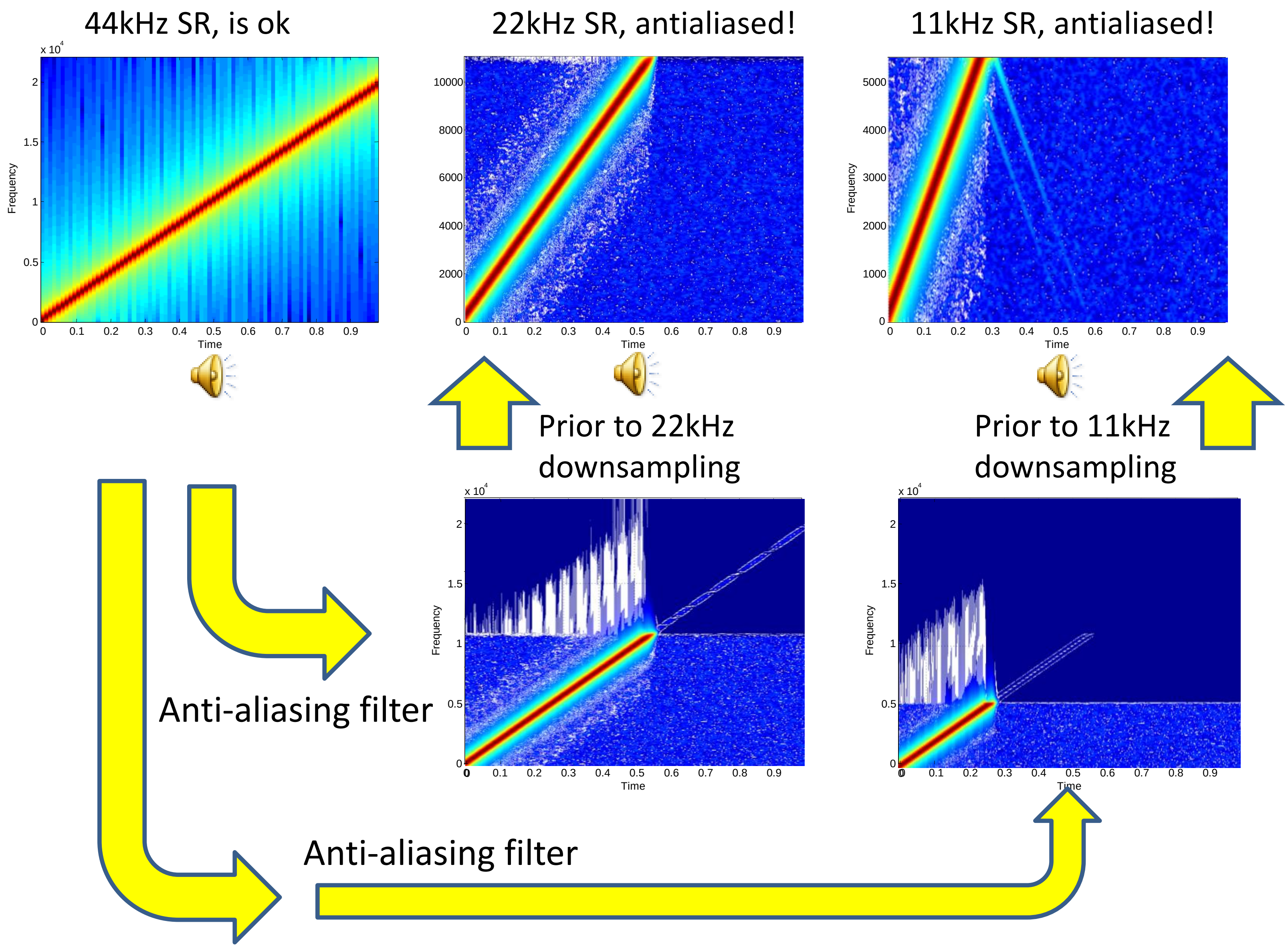
at 3kHz



Anti-Aliasing

- If a signal is sampled at *f* Hz, the highest frequency that will **not** be aliased is *f*/2.
  - Anti-aliasing is done to remove the frequencies above *f*/2 from the signal
  - Anti-aliasing filters are used for this

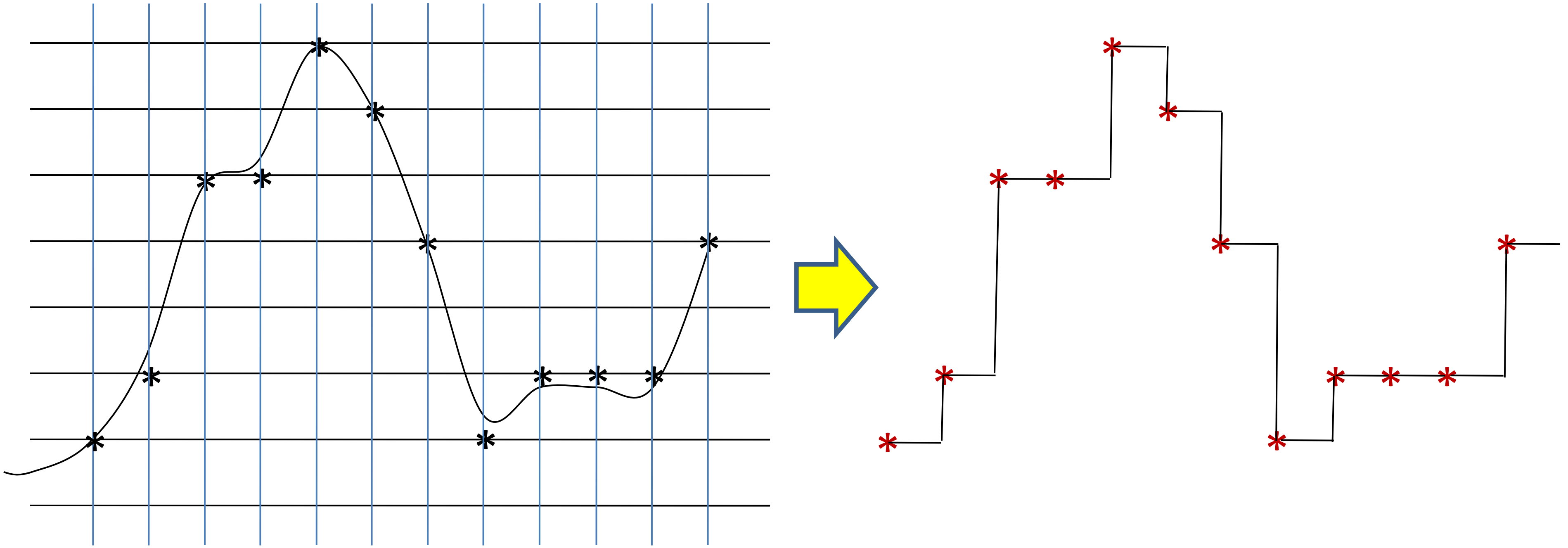
Anti Aliasing examples: Sweep 0-20khz



Typical Sampling Frequencies

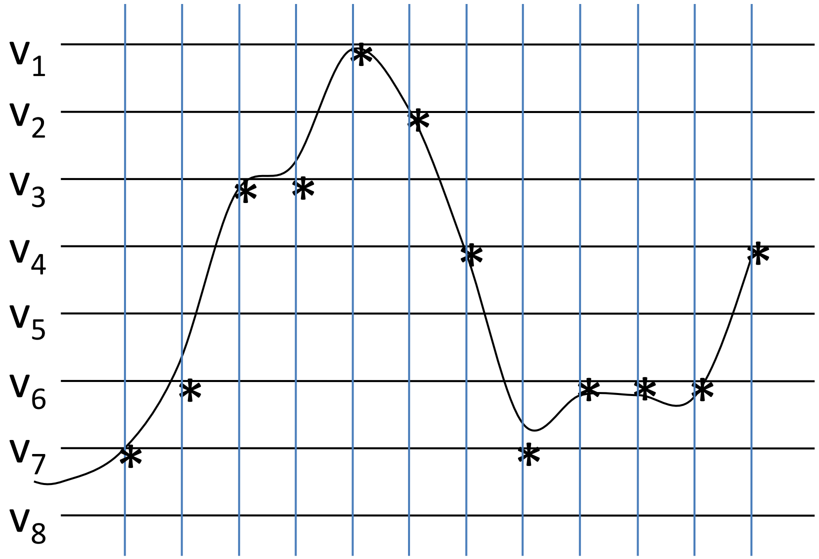
- Audio hardware in practical systems typically supports several standard rates
  - For speech 8kHz, 11.5kHz and 16kHz and 44.1 kHz are all used
  - Sampling hardware will usually employ appropriate anti-aliasing
    - Not always – be careful
  - CD recording employs 44.1 kHz per channel – high enough to represent most signals most faithfully
  - Telephone data is narrowband and has frequencies only up to 4 kHz
  - Good microphones provide a wideband speech signal. 16kHz sampling can represent audio frequencies up to 8 kHz. This is considered sufficient for capturing all speech content
    - Used for speech recognition, where possible

Sample Resolution



- Samples cannot take just any value
  - Each sample is represented by a limited number of bits
  - Sample values are usually integers, restricted to a finite set of values
    - Typically 2^N-1 integer values, for N = 8, 16, 32

Process of Quantization



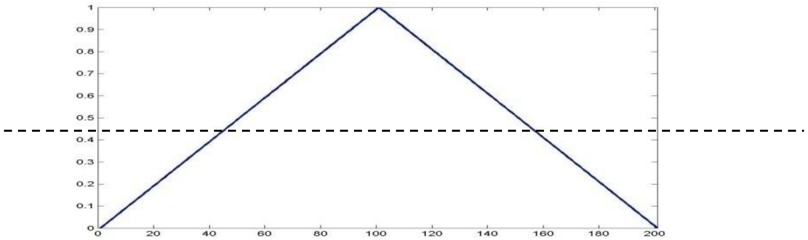
index	Value
0	v_1
1	v_2
2	v_3
3	v_4
4	v_5
5	v_6
6	v_7
7	v_8

- The allowed integer values can be used as indices
  - Mapping into the (pre-determined) values the signal will be rounded to
  - These are specified in a table of values
- **Only the indices are stored!**
  - Indices are the same as values if there is no additional quantization
  - **Table is important when any kind of nonlinear quantization is used**
    - The actual signal value is read off from the table

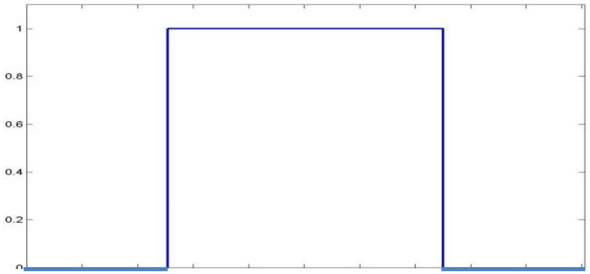
Mapping signals into bits

- Example of **1-bit** *Uniform* sampling (a.k.a Uniform quantization) table
  - Only 2 levels, using a threshold of 2.5 volts (this varies with hardware)

Signal Value	Bit sequence	Mapped to
S > 2.5v	1	1 * const
S <= 2.5v	0	0



Original Signal

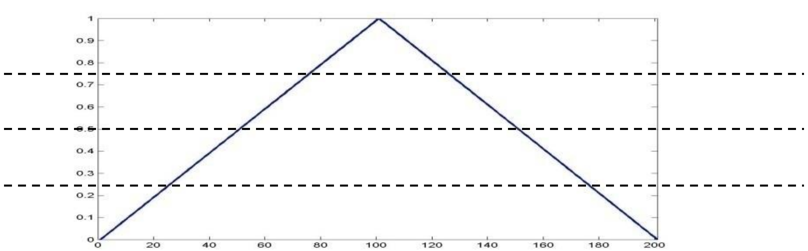


Quantized approximation

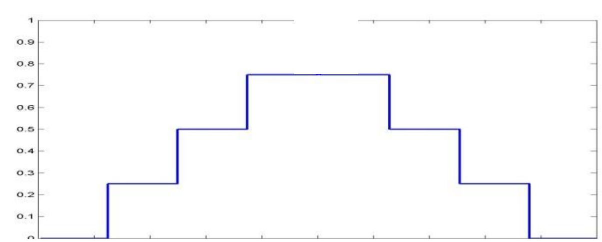
Mapping signals into bits

- Example of **2-bit** *Uniform* sampling table:
  - 4 equally-spaced levels, using example voltage thresholds

Signal Value	Bit sequence	Mapped to
S >= 3.75v	11	3 * const
3.75v > S >= 2.5v	10	2 * const
2.5v > S >= 1.25v	01	1 * const
1.25v > S >= 0v	00	0



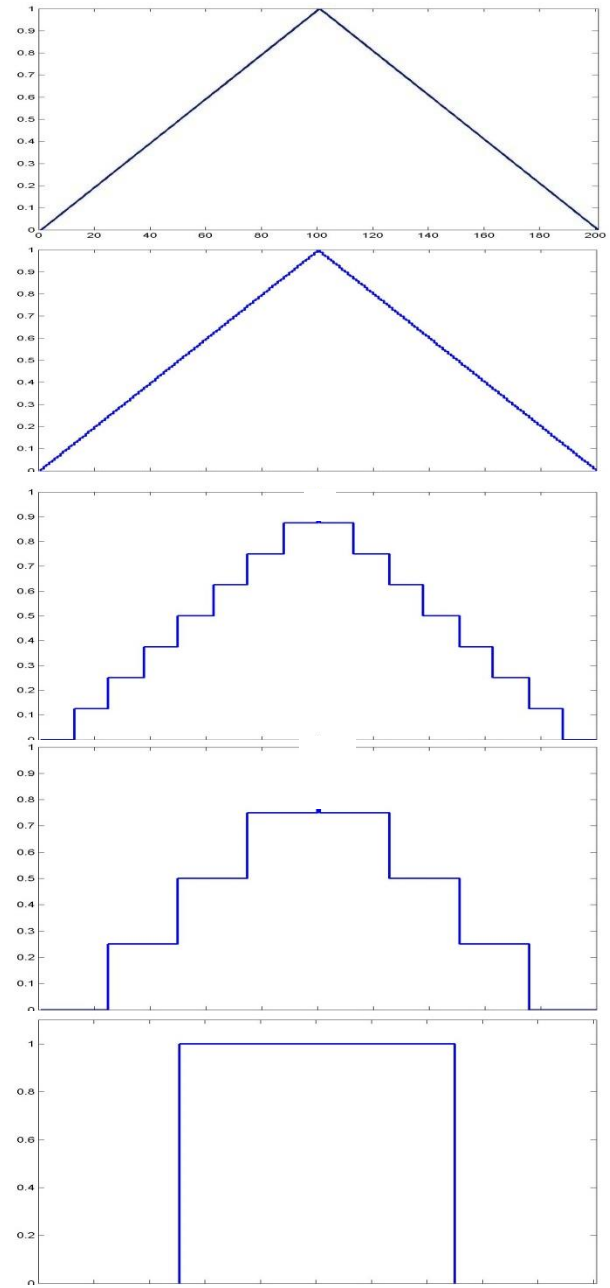
Original Signal



Quantized approximation

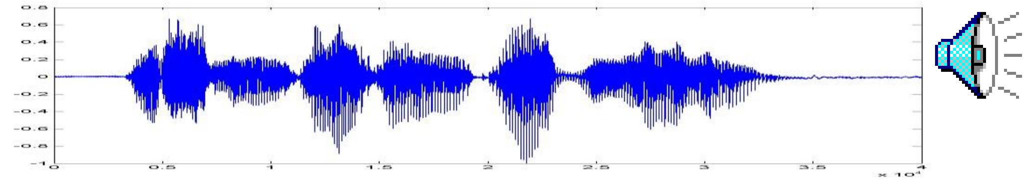
Uniform Sampling, different levels

- The original signal
- 8 bit quantization
- 3 bit quantization
- 2 bit quantization
- 1 bit quantization

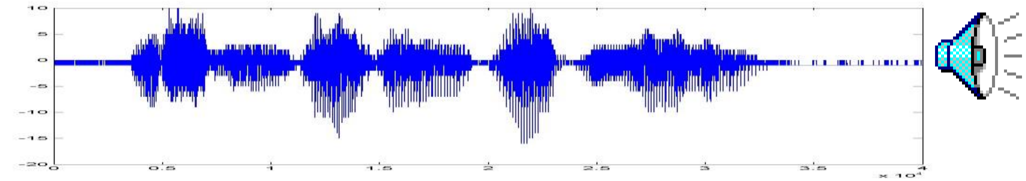


Tom Sullivan Says his Name

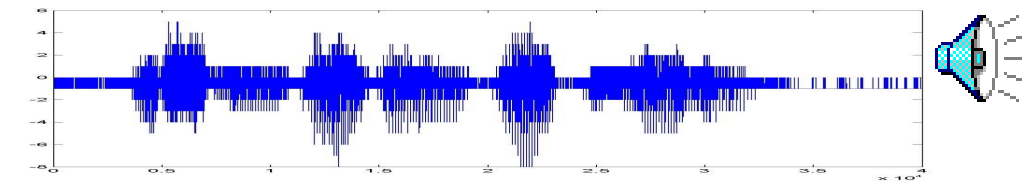
- 16 bit sampling



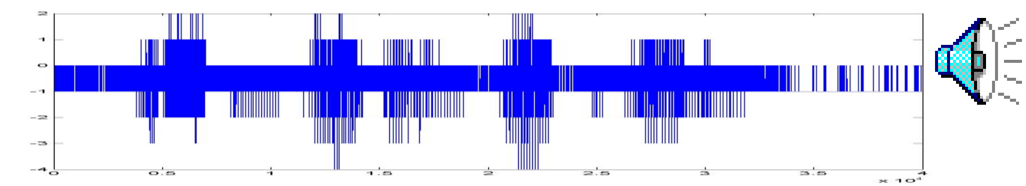
- 5 bit sampling



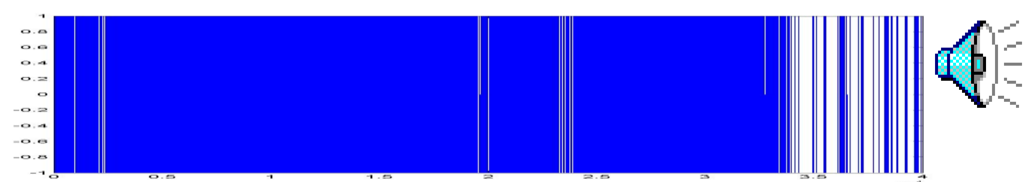
- 4 bit sampling



- 3 bit sampling

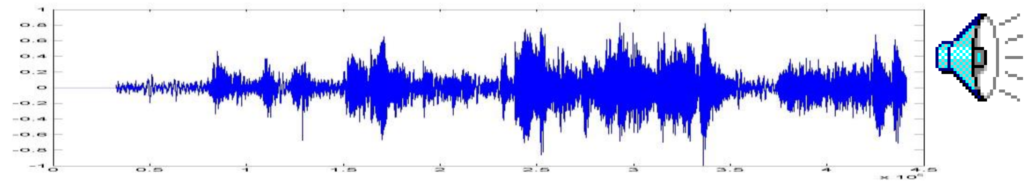


- 1 bit sampling

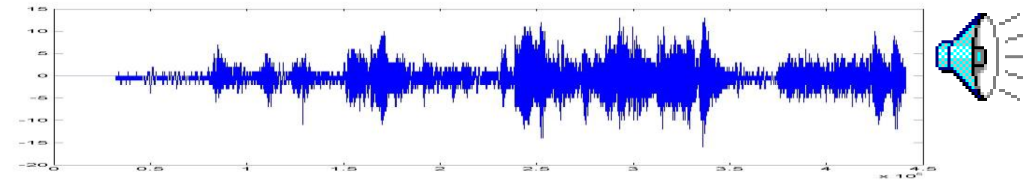


A Schubert Piece

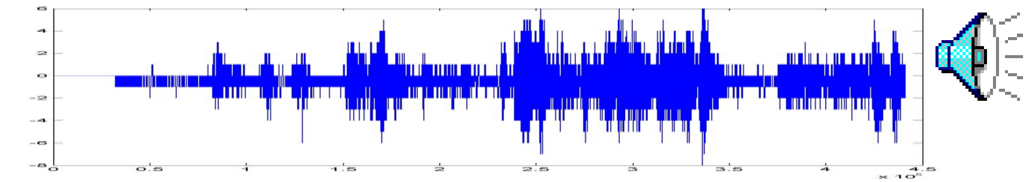
- 16 bit sampling



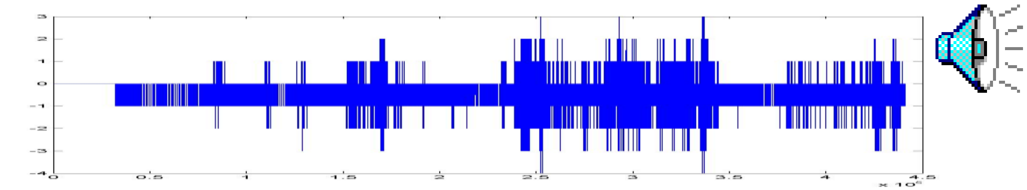
- 5 bit sampling



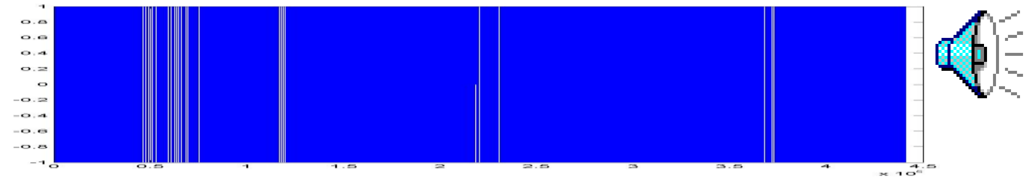
- 4 bit sampling



- 3 bit sampling

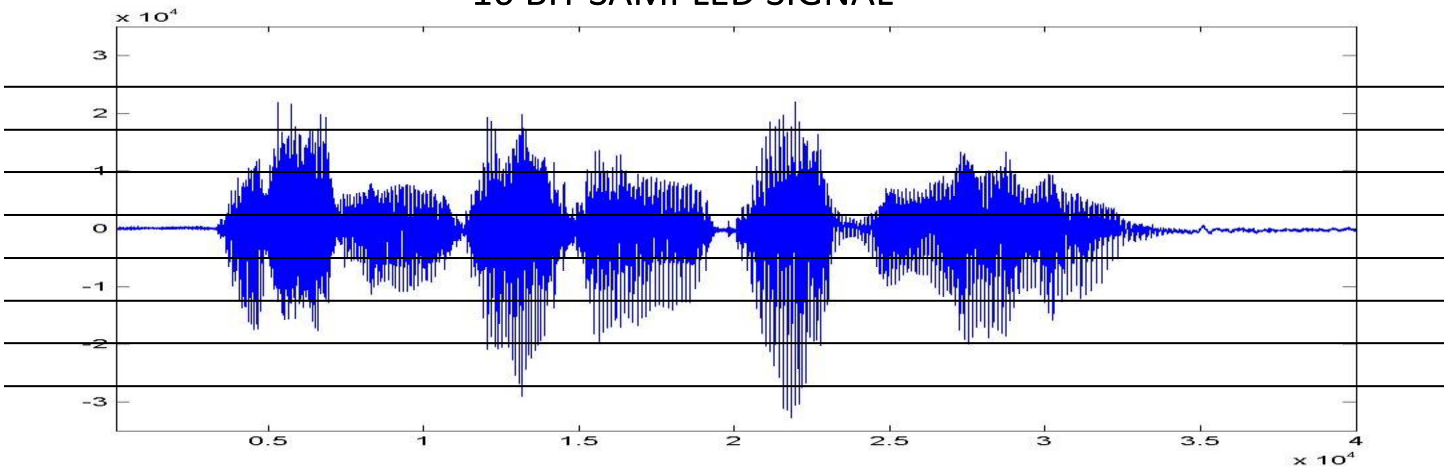


- 1 bit sampling

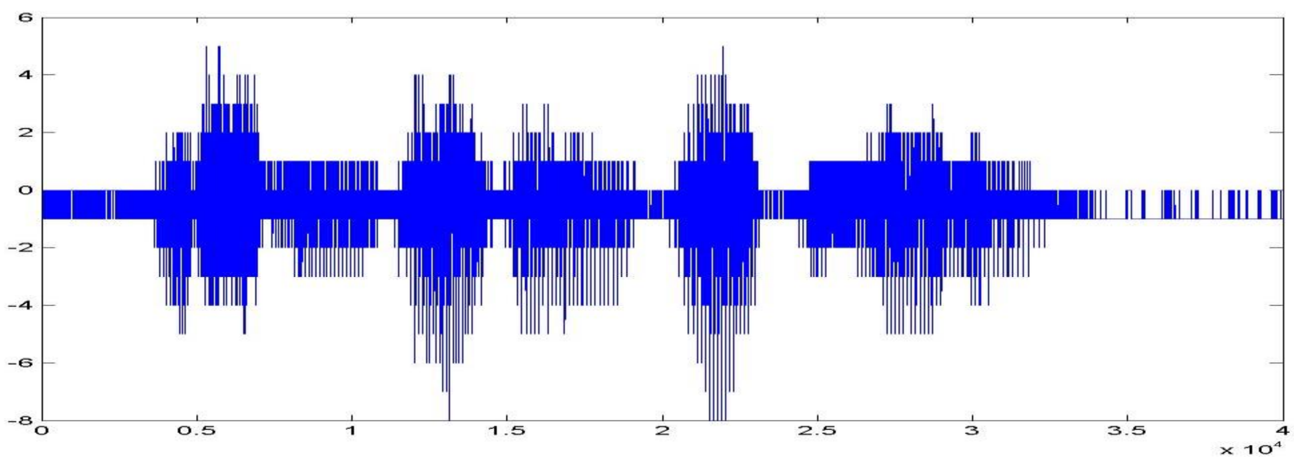


Improving on Uniform Sampling

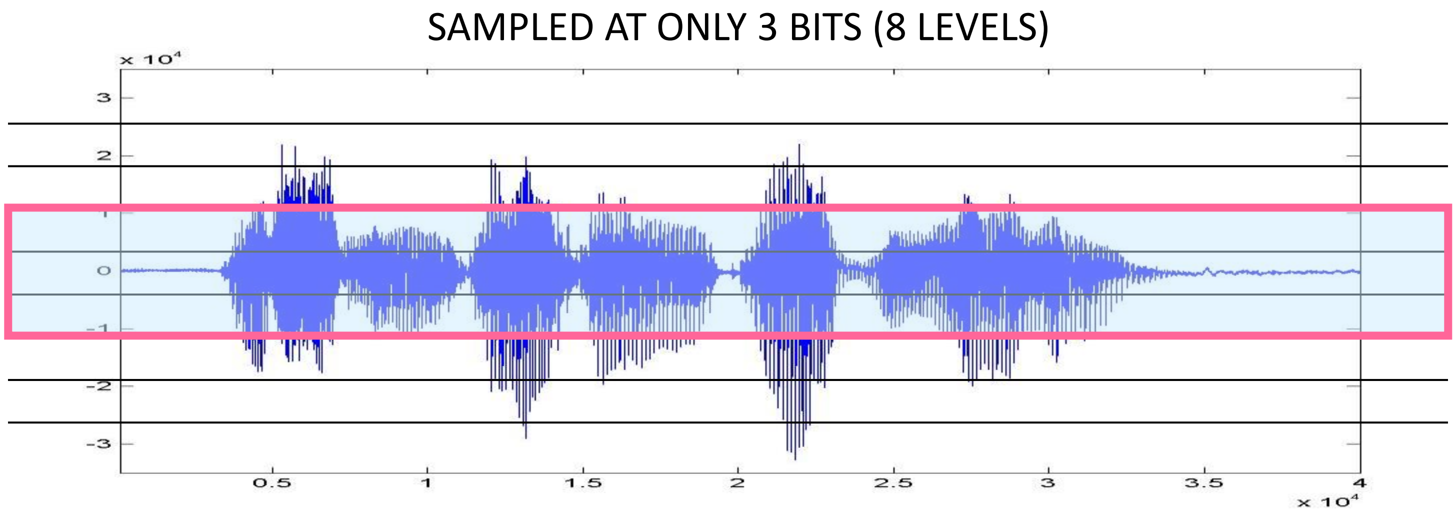
16 BIT SAMPLED SIGNAL



SAMPLED AT ONLY 3 BITS (8 LEVELS)



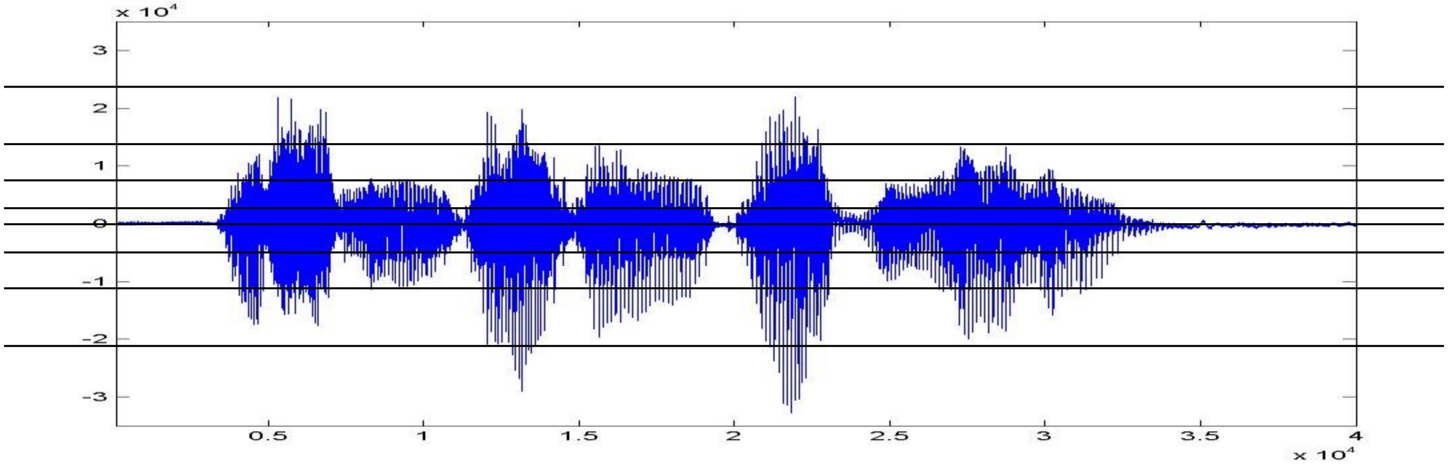
Improving on Uniform Sampling



- There is a lot more action in the central region than outside.
- Assigning only four levels to the busy central region and four entire levels to the sparse outer region is inefficient

Improving on Uniform Sampling

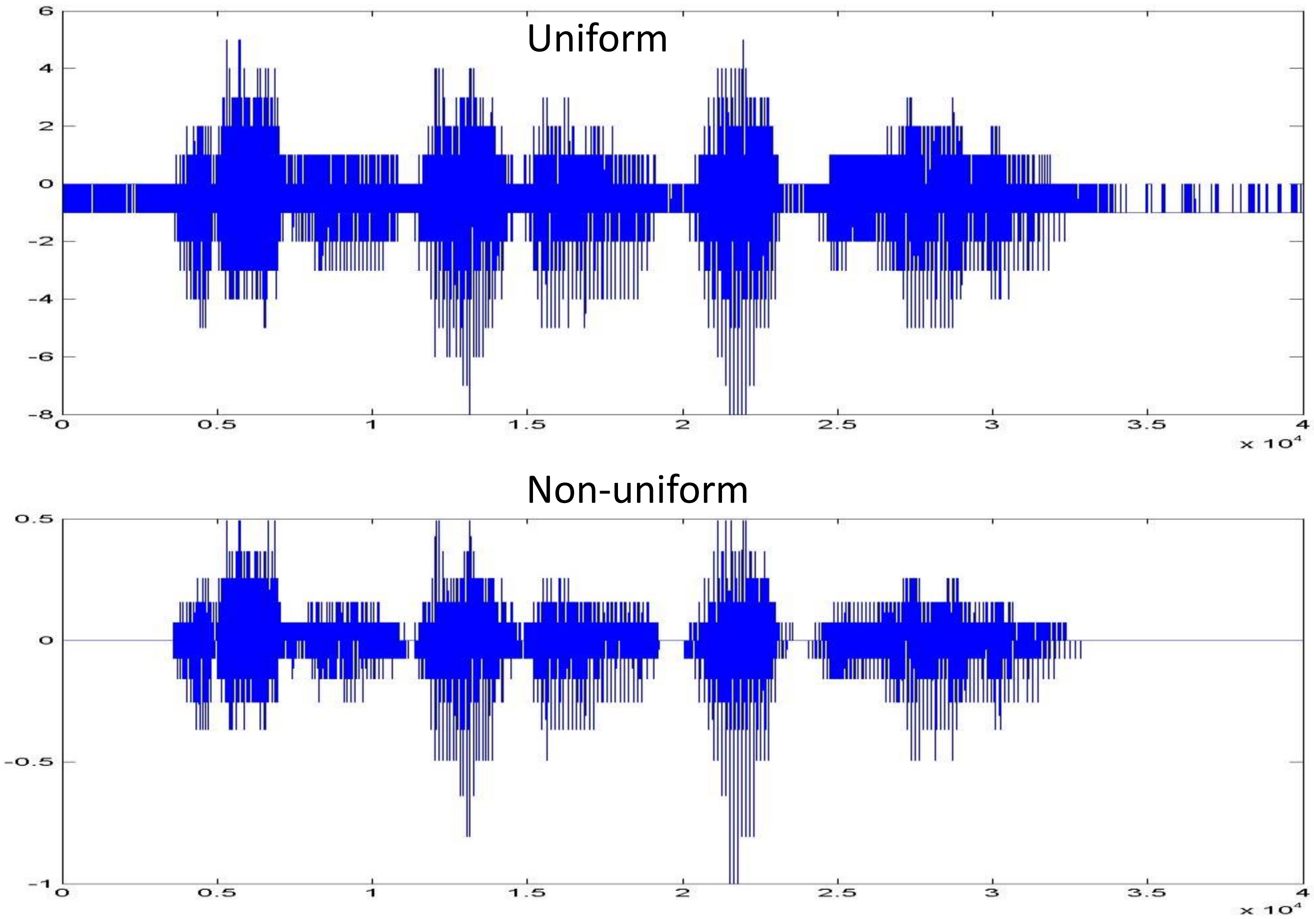
SAMPLED AT ONLY 3 BITS (8 LEVELS)



- Assigning more levels to the central region and less to the outer region can give better fidelity
  - for the same overall number of levels

Uniform vs. Non-uniform sampling

SAMPLED AT ONLY 3 BITS (8 LEVELS)

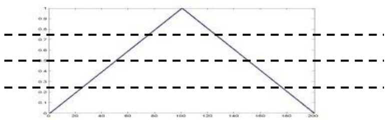


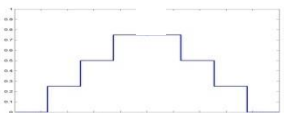
- Assigning more levels to the central region and less to the outer region can give better fidelity for the same storage

Sample Formats

Uniform sampling (quantization): Sample values equally spaced out

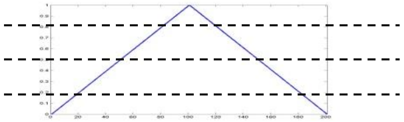
Signal Value	Bits	Mapped to
S >= 3.75v	11	3 * const
3.75v > S >= 2.5v	10	2 * const
2.5v > S >= 1.25v	01	1 * const
1.25v > S >= 0v	00	0

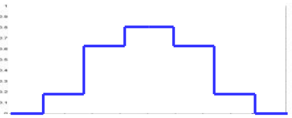




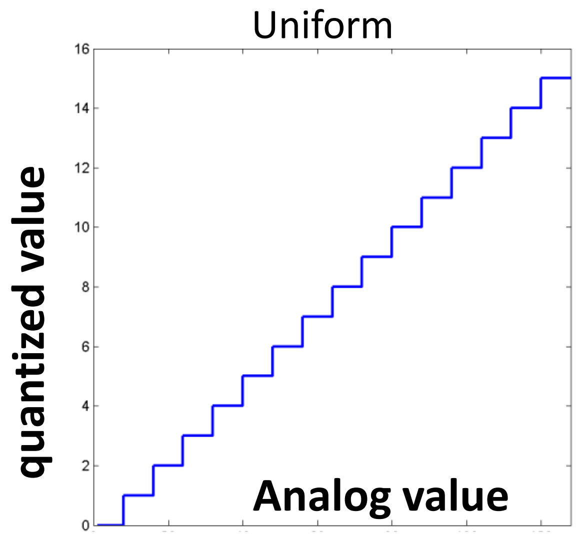
Non-uniform sampling

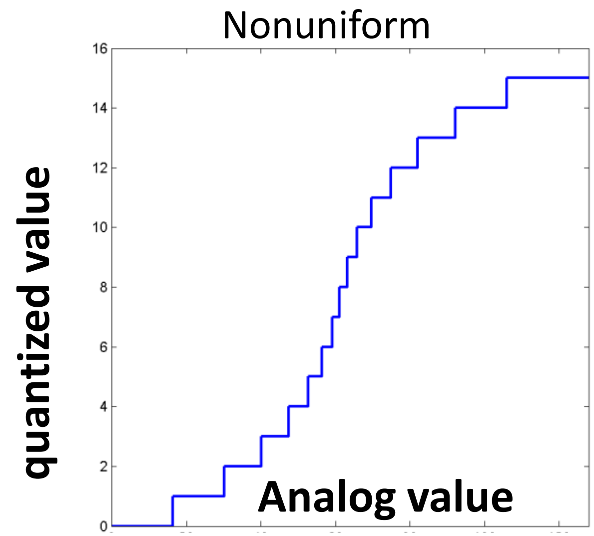
Signal Value	Bits	Mapped to
S >= 4v	11	4.5 * const
4v > S >= 2.5v	10	3.25 * const
2.5v > S >= 1v	01	1.25 * const
1.0v > S >= 0v	00	0.5 * const





Uniform vs. Non-uniform sampling





- Uniform sampling maps uniform widths of the analog signal to units steps of the quantized signal
- In non-uniform sampling the step sizes are smaller near 0 and wider farther away
  - The curve that the steps are drawn on follow a logarithmic law:
    - Mu-Law: $Y = C. \log(1 + \mu X/C)/(1+\mu)$
    - A-Law: $Y = C. (1 + \log(a.X)/C)/(1+a)$
- One can get the same perceptual effect with 8 bits of non-linear sampling as 12 bits of linear sampling

Capturing / Reading Audio

- The numbers returned by an ADC (analog-to-digital converter) or read from a file are *indices* into the table
  - The indices must be converted back to actual values
- For uniformly sampled data the indices are proportional to the value and can be used directly
  - Called "Linear PCM" or "PCM" encoding
- For non-uniform sampling, table lookup (or function inversion) is required
  - Often performed automatically by the audio data read / capture functions in most audio libraries

Audio capture: Conversion summary

Signal Value	Bits	Mapped to
S >= 3.75v	11	3
3.75v > S >= 2.5v	10	2
2.5v > S >= 1.25v	01	1
1.25v > S >= 0v	00	0

Signal Value	Bits	Mapped to
S >= 4v	11	4.5
4v > S >= 2.5v	10	3.25
2.5v > S >= 1v	01	1.25
1.0v > S >= 0v	00	0.5

- Capture / read audio in the format provided by the file or hardware
  - Linear PCM, Mu-law, A-law, Coded, others…
- Convert to 16-bit PCM value
  - i.e. map the bits onto the number on the right column
  - This mapping is typically provided by a table computed from the sample compression function
  - No lookup for data stored in PCM

Storing Audio/Speech Files

- The data are typically written in binary, but many of these formats have headers that can be read as ascii text.
  - Headers store critical information such as byte order, no. of samples, coding type, bits per sample, sampling rate etc.
  - "bit depth": how many bits are used to represent each sample

- Audio files must be converted from stored format to linear PCM format for further processing
  - Audio I/O library routines will usually process the headers to obtain the necessary information and perform the appropriate conversion. We can also write the I/O ourselves.

- **There are many storage formats in use. 3 categories of audio formats:**
  - **Uncompressed**
  - **Lossy Compressed**
  - **Lossless Compressed**

Audio file formats and codecs

UNCOMPRESSED FORMATS

- Digitized signal stored without further processing, take up a lot of storage space
  - E.g. 34 MB per minute for 24-bit 96 KHz stereo.

PCM (“raw” data, *.raw)

- PCM (Pulse-Code Modulation): Analog sound waves sampled at regular intervals (pulses) and stored. This term is interchangeably used with LPCM (Linear Pulse-Code Modulation) in which samples are taken at linear intervals
- Commonly used in CDs and DVDs

NIST (*.sph)

  - container
  - 1024 byte ascii header, followed by PCM format

SUN (*.au, *.snd)

  - **Legacy**

Audio file formats and codecs

WAV (*.wav)

- **Microsoft PCM and ADPCM:** “windows” audio.
  - Standard for this was developed by Microsoft and IBM in 1991.
  - In PCM, data for .WAV files is stored using linear samples 8 bits per sample, while ADPCM uses deltas between samples at 4 bits per sample
  - ADPCM (Adaptive Delta (or Differential) Pulse Code Modulation)
    - ADPCM takes up half the disk space as PCM. Suitable for longer files
    - It stores the value differences between two adjacent PCM samples and makes some assumptions that allow data reduction. Because of these assumptions, low frequencies are properly reproduced, but high frequencies tend to get distorted. The distortion is easily audible in 11 kHz ADPCM files

- WAV: Waveform Audio V?
  - Windows container for audio formats
  - Made suitable for windows o/s. Mac o/s supports it as well
  - WAV files can contain both compressed and uncompressed audio, but mostly always contain uncompressed audio in PCM format

Audio file formats and codecs

AIFF

- AIFF (Audio Interchange File Format)
  - Developed by Apple for Mac systems in 1988
  - Also a container
  - Can contain multiple kinds of audio, e.g. AIFF-C (contains compressed audio), older version called Apple Loops which is used by GarageBand and Logic Audio etc.
  - All use the same aiff file extension.
  - Most AIFF files contain uncompressed audio in PCM format. Windows systems can also handle them.

Audio file formats and codecs

LOSSY COMPRESSION

- Some information is lost, mostly imperceptible to the human ear
- Too much or too often compressed audio can have perceptual artifacts

MP3 (*.mp3)

- MP3 (MPEG-1 Audio Layer 3, MPEG-2 Audio Layer 3, NOT THE SAME AS MPEG-3)
  - MPEG-1 and MPEG-2 are multimedia containers
  - MPEG-3 : A container. Was designed to handle HDTV signals at 1080p in the range of 20 to 40 megabits per second
- Introduced in 1993. Popular for music. Almost all devices can handle it today
- Based on perceptual coding. Removes frequencies that are outside of hearing capabilities of people (decided based on psychoacoustic principles)
  - Reduce the quality of sounds that are not easy to hear, compress the important parts of the audio as efficiently as possible

Audio file formats and codecs

AAC (*.aac)

- AAC (Advanced Audio Coding).
- Developed in 1997 as the successor to MP3. The compression algorithm is more advanced. Codec is more efficient. At he same bitrate, AAC has better sound quality than MP3
- Standard audio compression method used by YouTube, Android, iOS, iTunes, PlayStations etc.

OGG (Vorbis)

- A multimedia container that can hold all kinds of compression formats, but is most commonly used to hold Vorbis files. Vorbis was first released in 2000. Opensource. Better than most other lossy compression formats (smaller file size for equivalent audio quality). Good for mid to high quality (8kHz-48.0kHz, 16+ bit, polyphonic) audio and music at fixed and variable bitrates from 16 to 128 kbps/channel)
- **Vorbis** is a free and open-source software project headed by the Xiph.Org Foundation. The project produced Ogg.

WMA

- WMA (Windows Media Audio). Released in 1999. Proprietary to Microsoft
- Developed to address flaws in the MP3 compression.
- Compression algorithm quite similar to AAC and OGG. Better quality than MP3
- Not many devices/platforms support it since it is proprietary

Audio file formats and codecs

LOSSLESS COMPRESSION

- Compresses without any loss from original source.
- Not as efficient as lossy compression. Equivalent files can be 2x to 5x larger
- Do not confuse lossless compression with high-resolution audio (which is most likely a scam)

FLAC

- FLAC (Free Lossless Audio Codec). Developed in 2001.
- Can compress up to 60% without losing a single bit of information. Open source.
- Alternative to MP3 for CD audio.
- Full quality of raw uncompressed audio in half the file size

Audio file formats and codecs

ALAC

- ALAC (Apple Lossless Audio Codec). Developed in 2004. Proprietary initially, became opensource in 2011
- less efficient than FLAC. But Apple users are forced to use it because iTunes and iOS support ALAC and do not support FLAC at all.

WMA

- WMA (Windows Media Audio). Lossless alternative to lossy WMA (Windows is seriously confused….) This version is called “WMA Lossless”. Used the same extension as the lossy version.
- Worse than FLAC and ALAC in compression efficiency. Proprietary. Supported by Windows and Mac o/s.

Editing audio

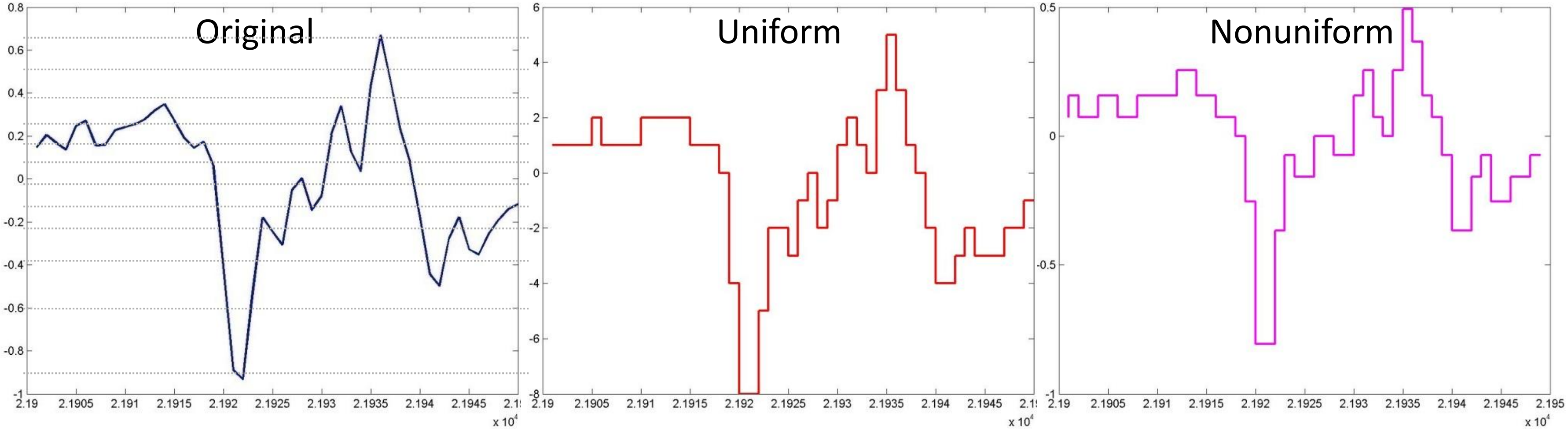
- Should be done on raw/uncompressed format

Signal Quality

- The **quality** of the digitized signal depends critically on many factors:
  - **The electronics performing sampling and digitization**
    - Poor quality electronics can severely degrade signal quality
      - *E.g.* Disk or memory bus activity can inject noise into the analog circuitry
  - **Anti-aliasing (proper/improper)**
    - Not using an anti-aliasing filter is a cause for many problems
  - **Quantization levels (sufficient/insufficient)**
    - Minimally 16 bit PCM or 8 bit Mu / A-law is needed
  - **Proper setting of the recording level**
    - Too low a level underutilizes available signal range, increasing susceptibility to noise
    - Too high a level can cause *clipping*
  - **The microphone quality**
  - **Ambient noise in recording environment**

- Suboptimal signal quality can affect analysis accuracy to the point of being completely useless
- **When signals of different quality are spliced together, the overall sound may have serious artifacts**

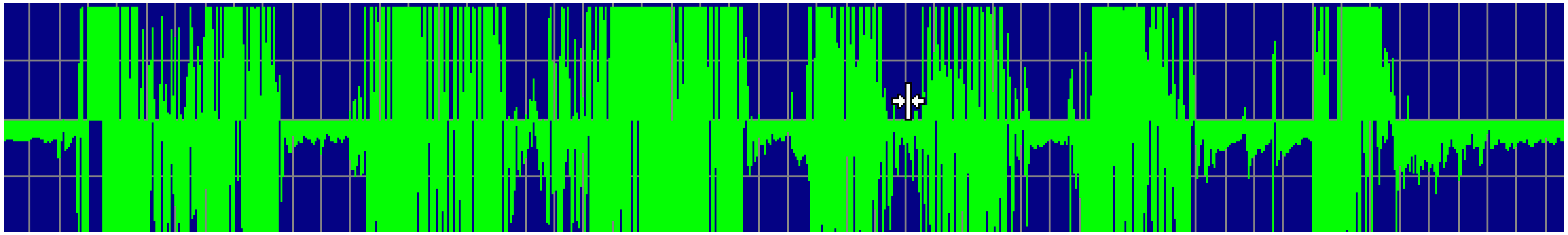
Clipping in audio signals



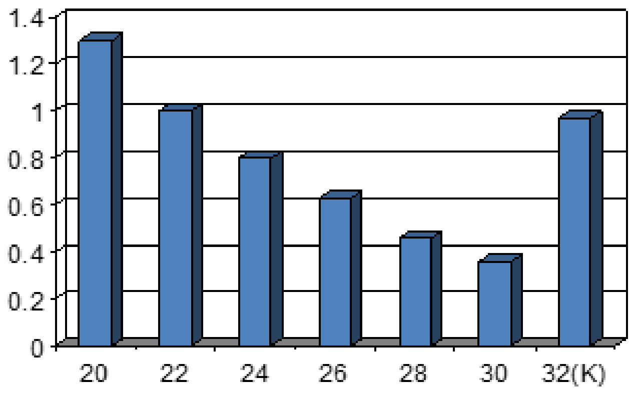
- At each sampling instant, the value of the waveform is rounded off to the nearest quantization level
- Samples outside the quantization range are given either the highest or lowest quantized values
  - E.g. max value for 8 bit quantization is 255. If sample = 300, it is set to 255
  - **This is called clipping**

Clipping

- Clipping and non-linear distortion are the most common problems in audio recordings
  - While recording, clipping can be fixed by reducing signal gain (but AGC is not good)

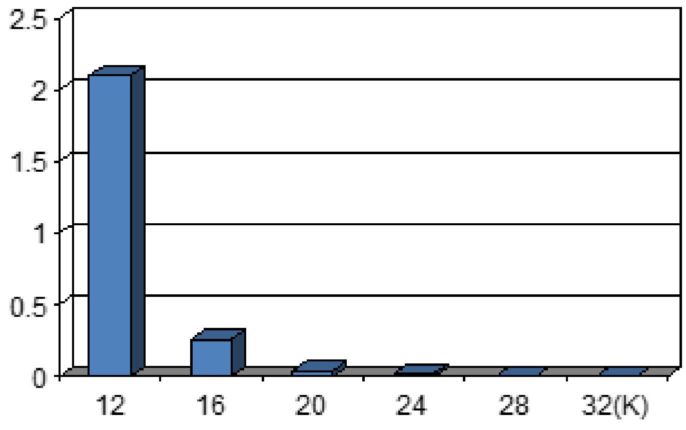


Clipped signal histogram



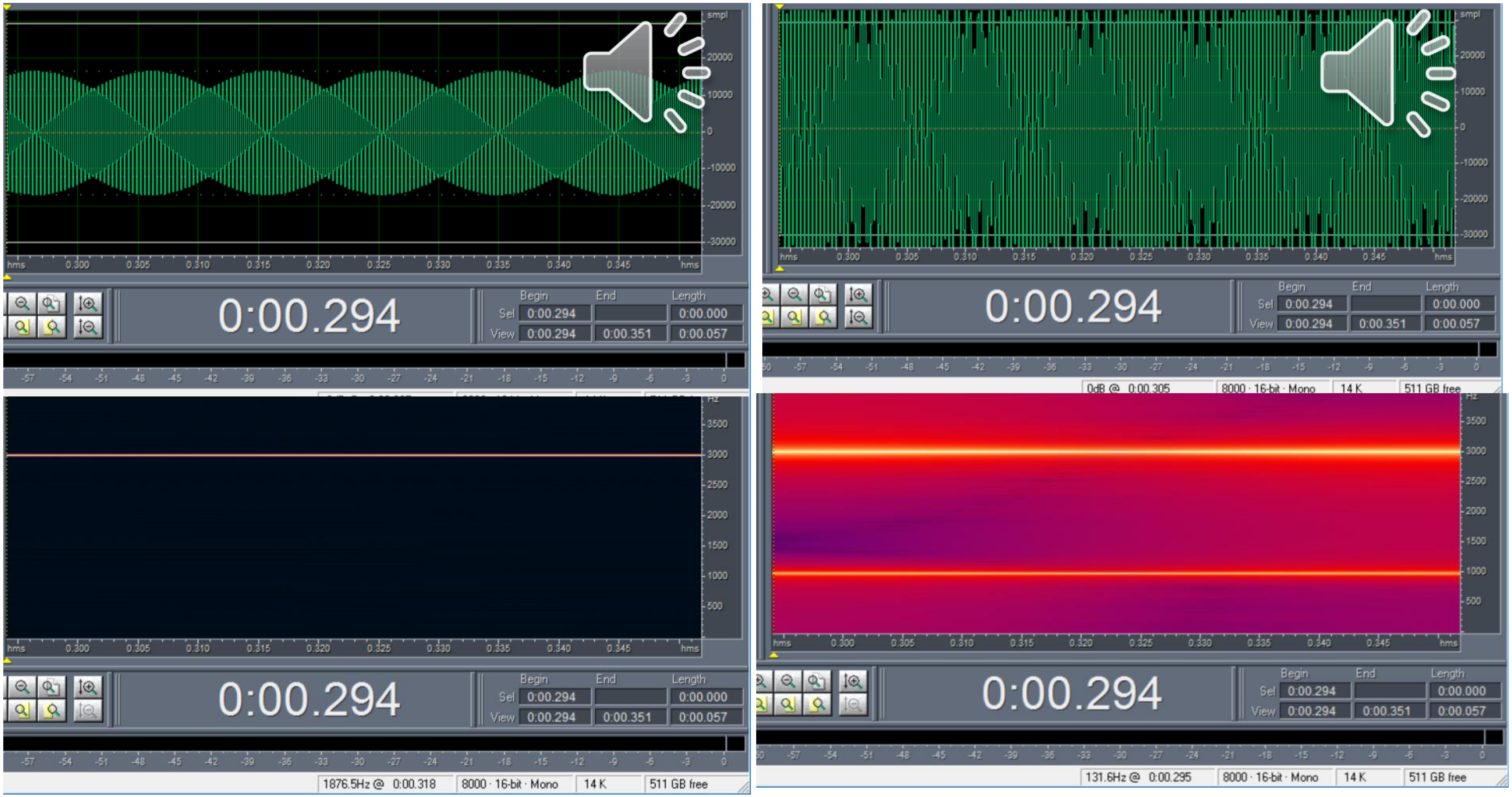
Absolute sample value

Normal signal histogram



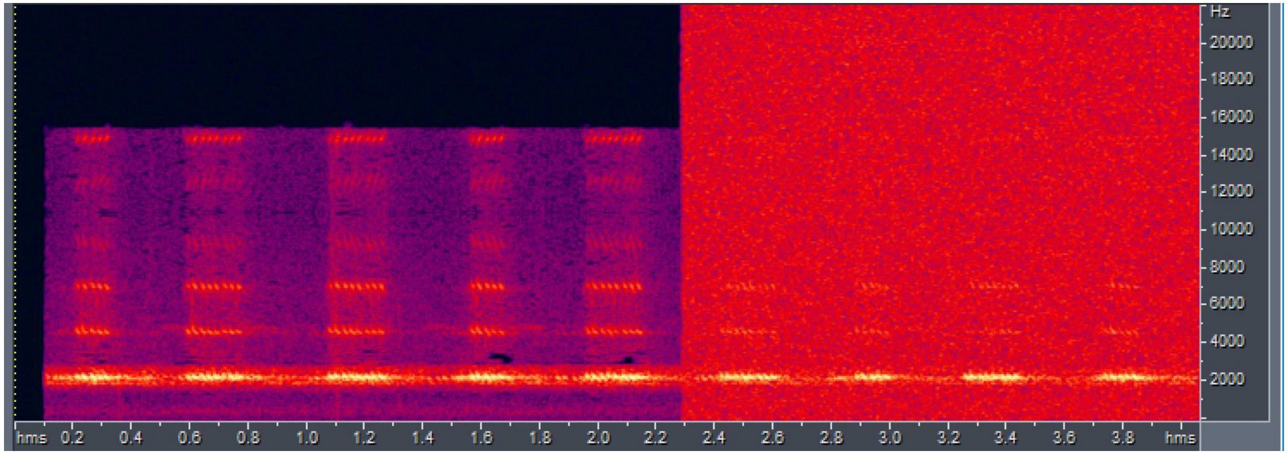
Absolute sample value

What does clipping do?



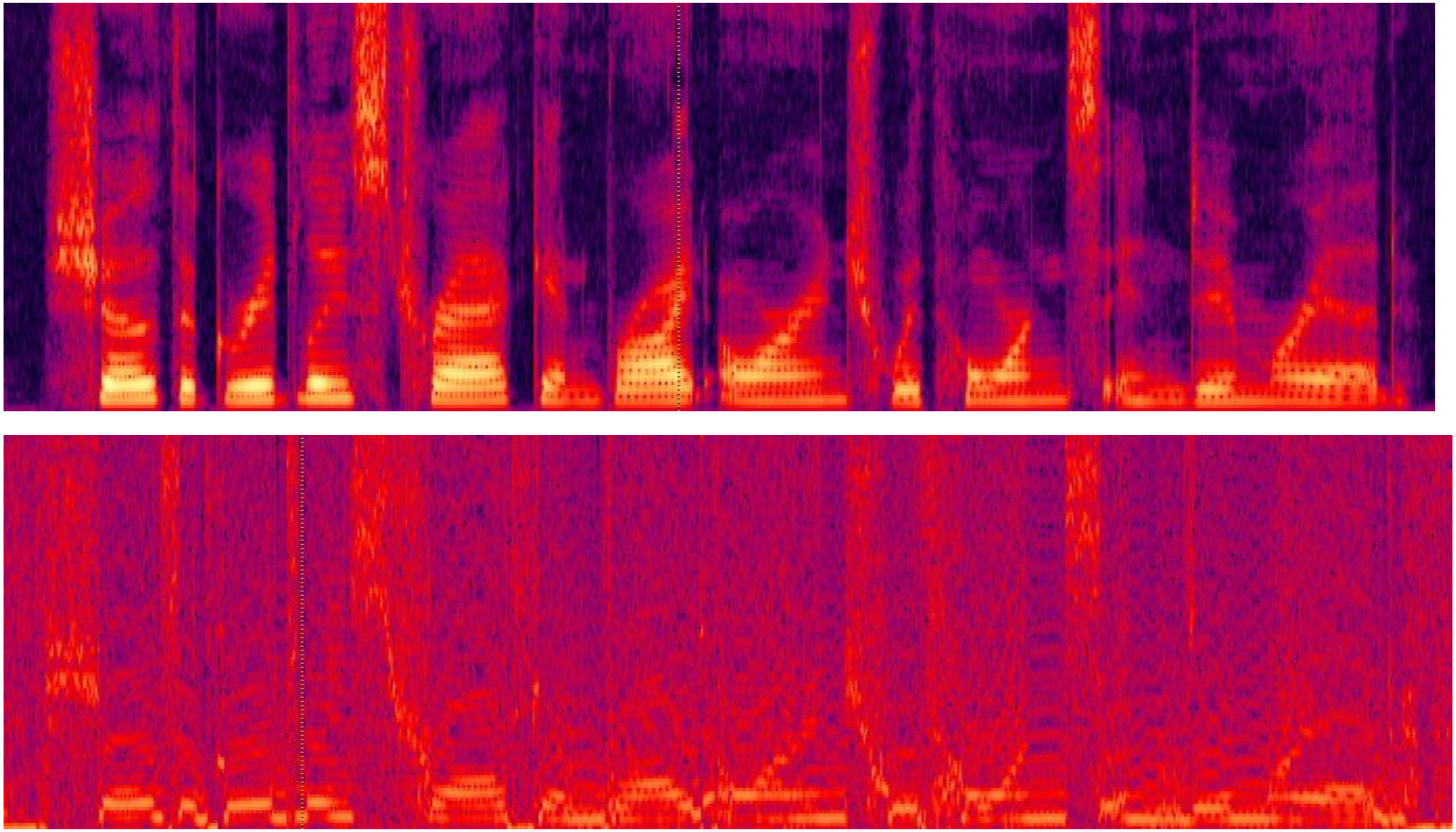
- It causes aliasing

Background Noise



- Signals in the background can corrupt the recording
  - Diffuse background noise, e.g. in an automobile
  - Localized sounds, e.g. air conditioner
  - Background talkers, music..
  - Denoising algorithms are used for cancelling noise, but they leave artifacts behind

Background Noise

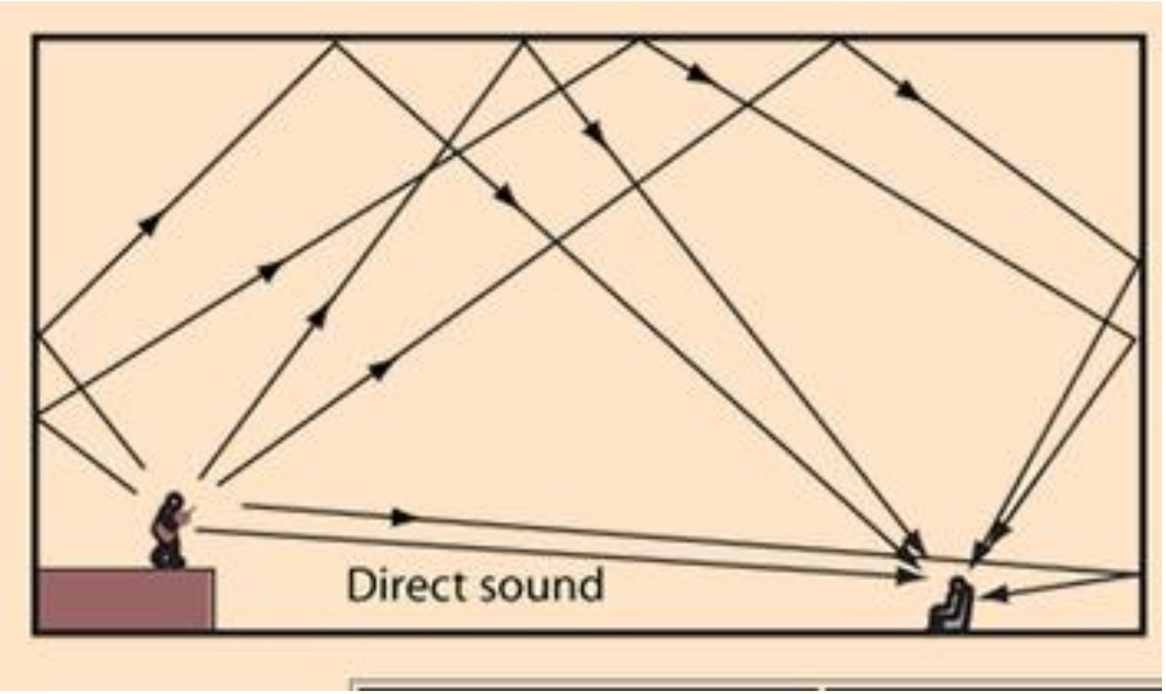


- Clean and noisy speech

Recording Channel

- Recording channels/media can result in signal distortions
  - Telephone channels reduce the bandwidth of the signal
    - 300-3300Hz
  - Cellphone / VOIP channels introduce *coding* distortion
    - Coding and decoding speech introduces distortions
  - They can also result in *discontinuities* from dropped packets
  - Poor microphones can result in spectral distortions
  - Computer recordings may be affected by memory and disk activity
- Distortions cannot usually be recovered from
- **Tampered signals may show anomalous patterns in distortion**

Reverberation



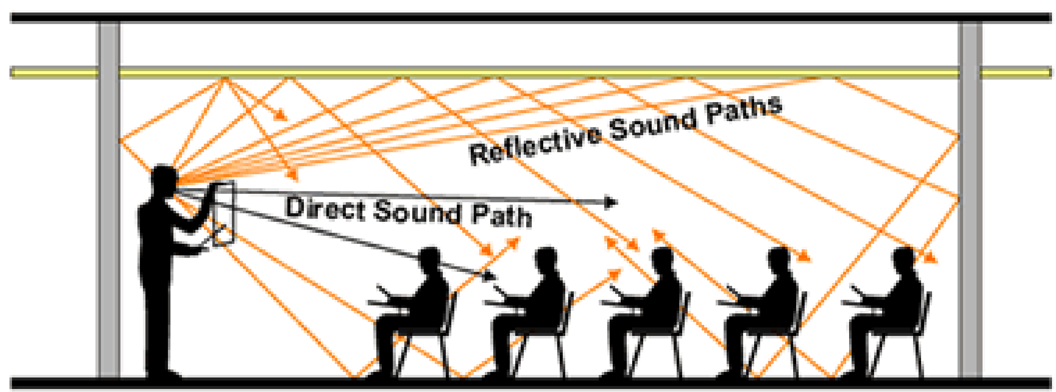
- Reflective surfaces (e.g. walls) can introduce *reverberation* into a recording
  - Due to repeated addition of a signal with delayed reflections of itself

Reverberation

REVERBERATION TIME

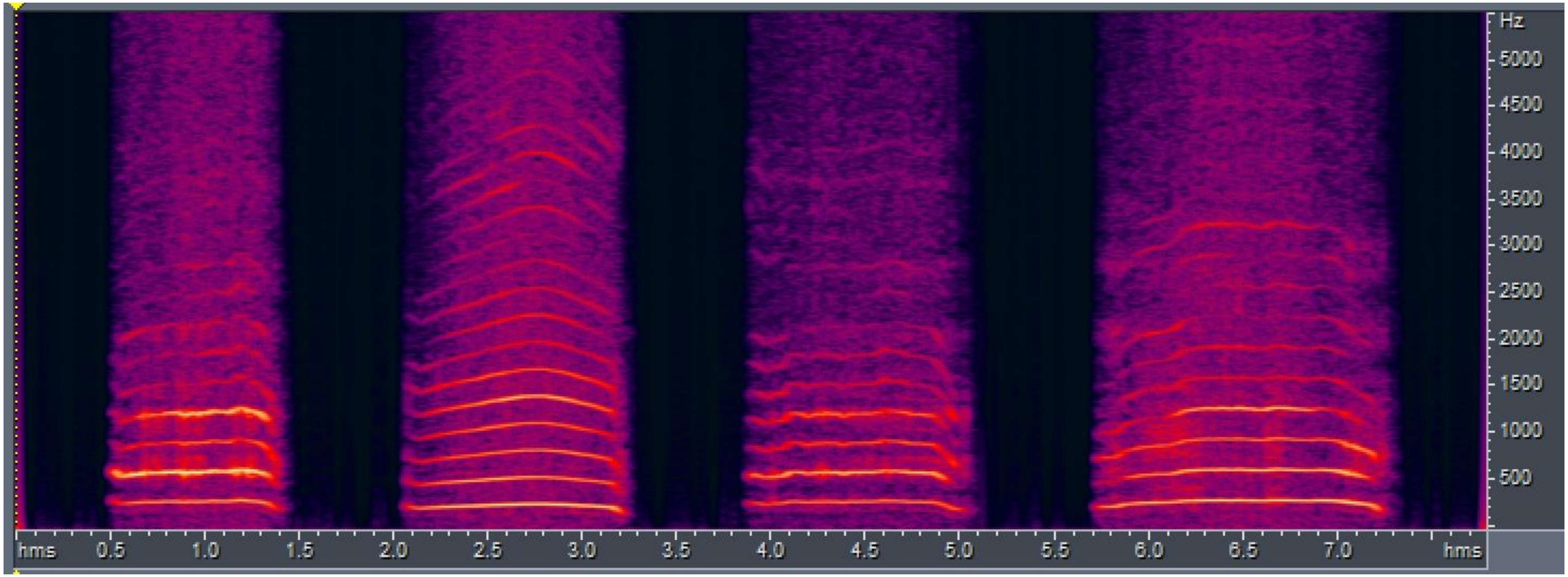
The time it takes for reflected sound to die down by 60 decibels from the cessation of the original sound signal (measured in seconds).

- Reflected sound tends to "build up" to a level louder than direct sound. Reflected sounds **MASK** direct sound.
- Late arriving reflections tend to **SMEAR** the direct sound signal.



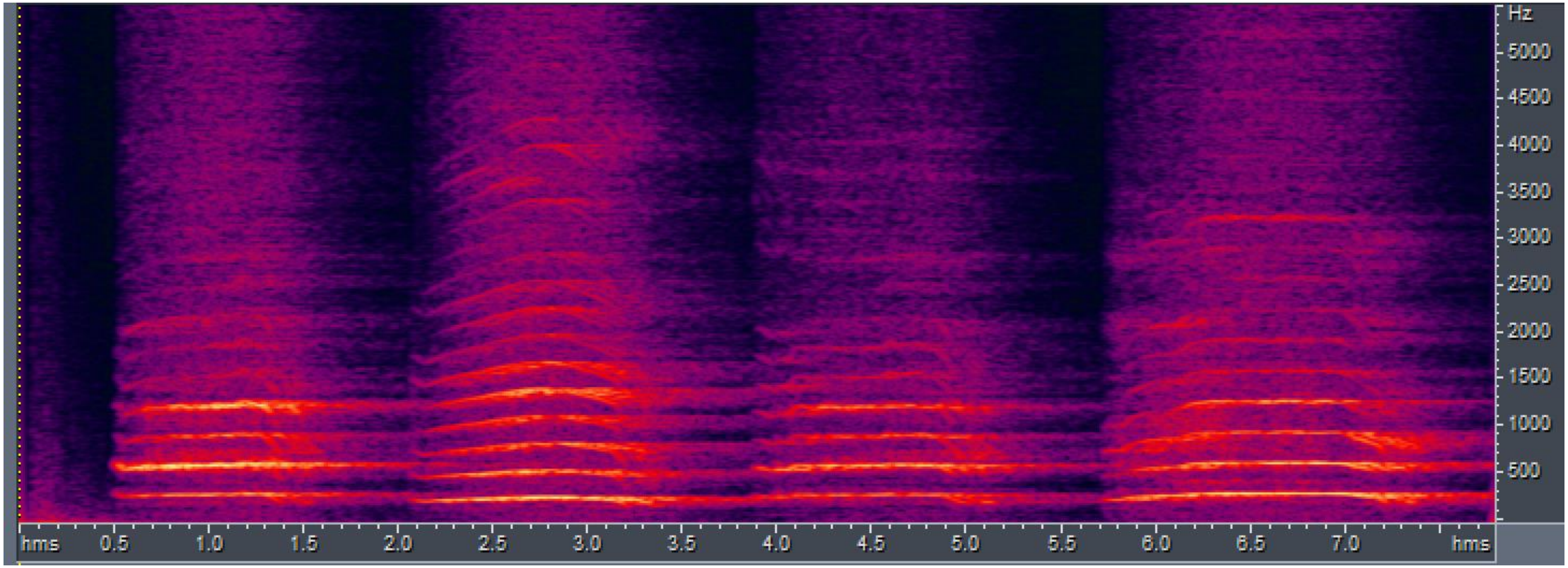
- Each person hears the sound slightly differently

Reverberation



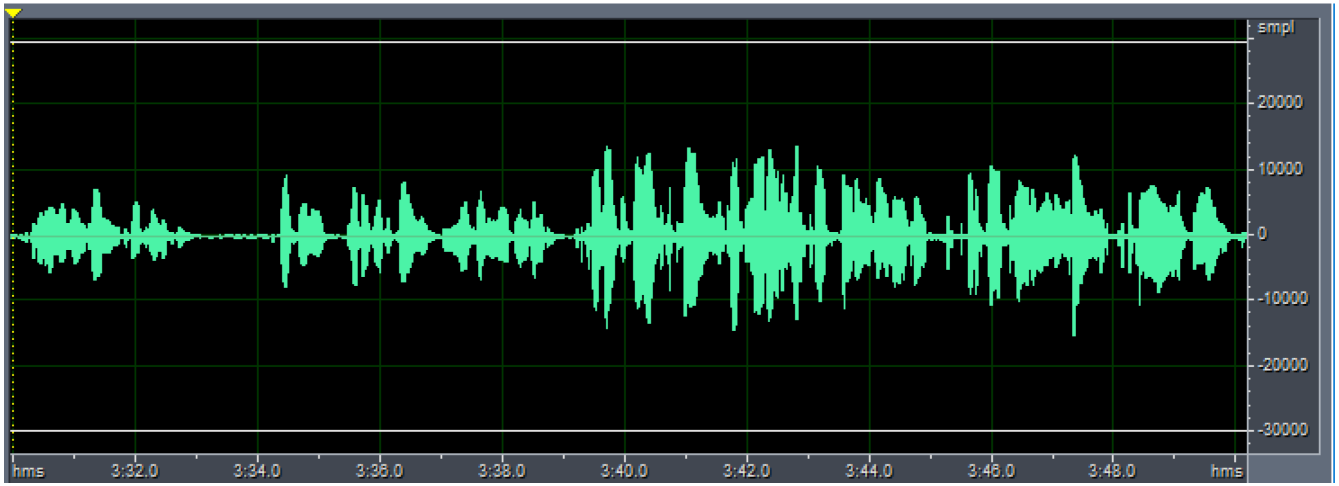
- Lama

Reverberation



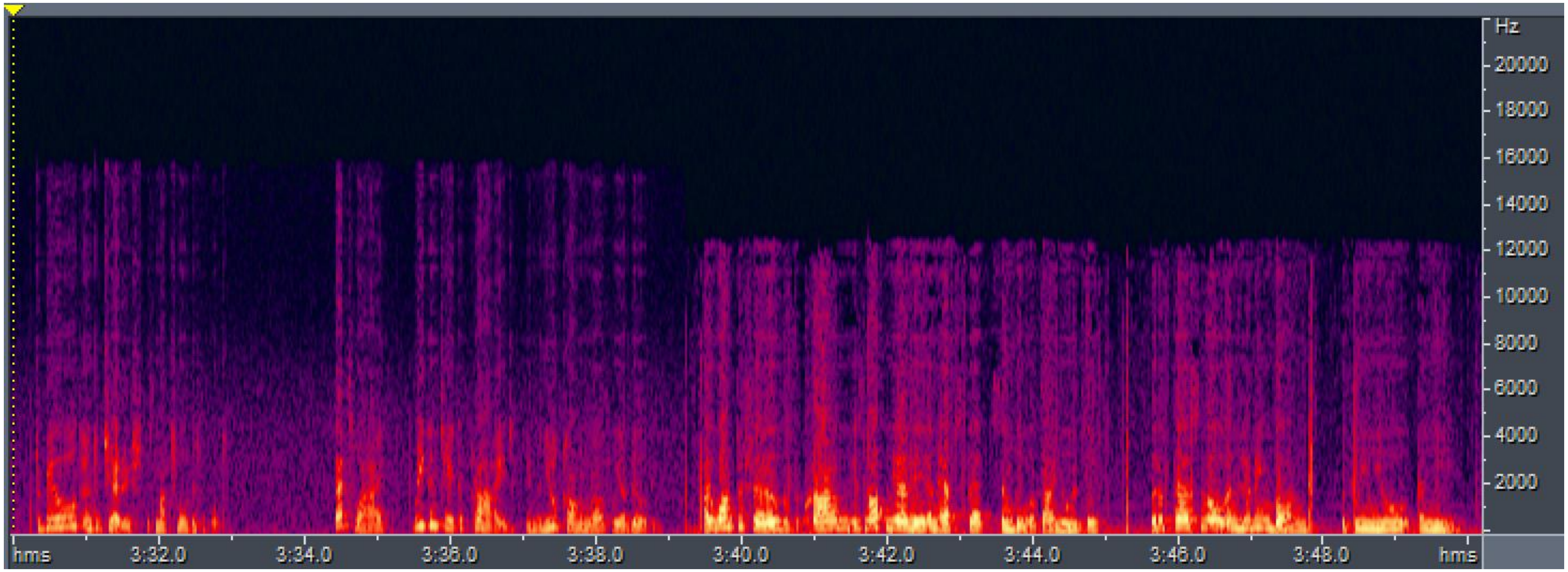
- Lama, reverb in large hall

Splicing boundary effect



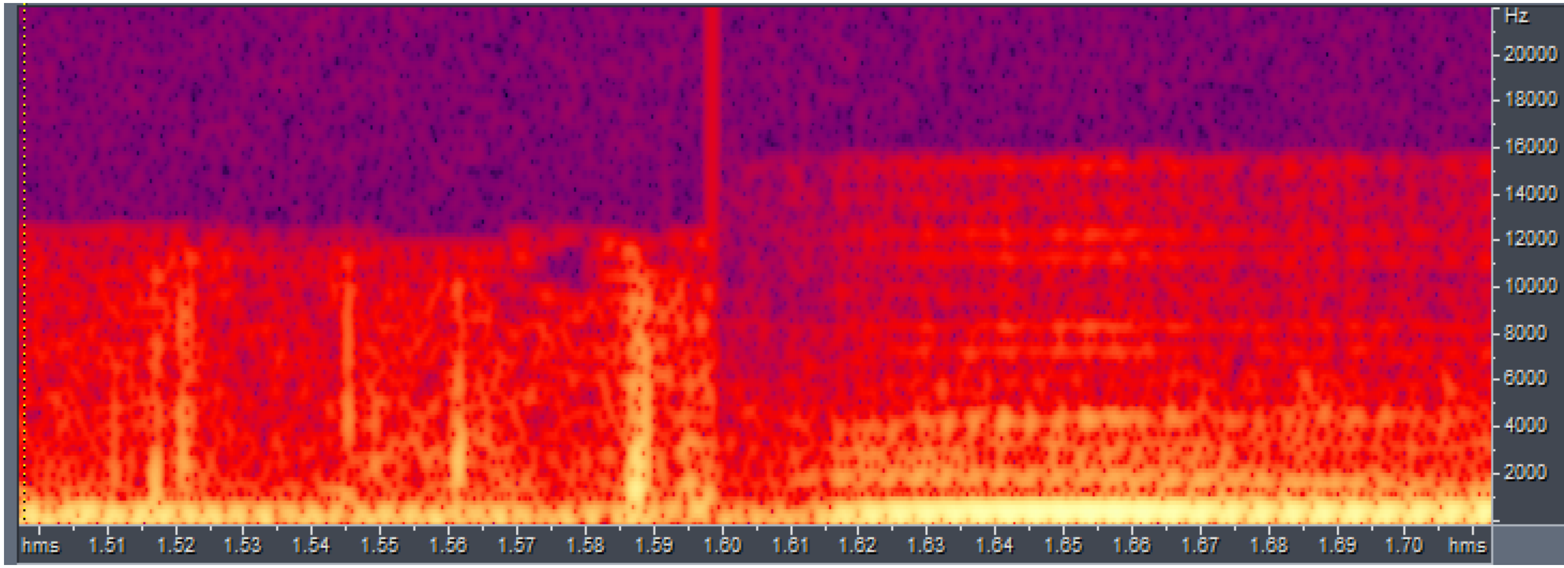
- The queen of England, 1953
- Tampered signal in time domain

Splicing boundary effect



- Snippets from two different speeches of the Queen of England in 1953

Splicing boundary effect



– The queen of England, 1953: “I want to sell the building”

In this lecture

- Digital multimedia: Recording and devices
  - Audio
  - Images
  - Video
  - Text
- Digital multimedia: Processing
  - Audio processing
  - Two generic processing techniques

In the next lecture

- Digital multimedia: Recording and devices
  - Audio
    - Audio processing
  - Images
  - Video
  - Text