AUDIO

ARIHARASUDHAN

Audio : A Feeling

There is a folklore in the southern part of Tamil Nadu that revolves around The Akka Kuyil (Common Hawk Cuckoo). According to the tale, The Akka Kuyil witnesses her sister being swept away by a flood in the village brook. Overwhelmed with sorrow, The Akka Kuyil becomes heartbroken. Every monsoon season in the south, The Akka Kuyil calls out to her sister whenever it encounters the brook, hoping for her sister's return. This emotional story resonates deeply with the people of the region, including myself. The Akka Kuyil's voice is so sympathetic. Audio is a feeling.

Sound : A Phenomenon

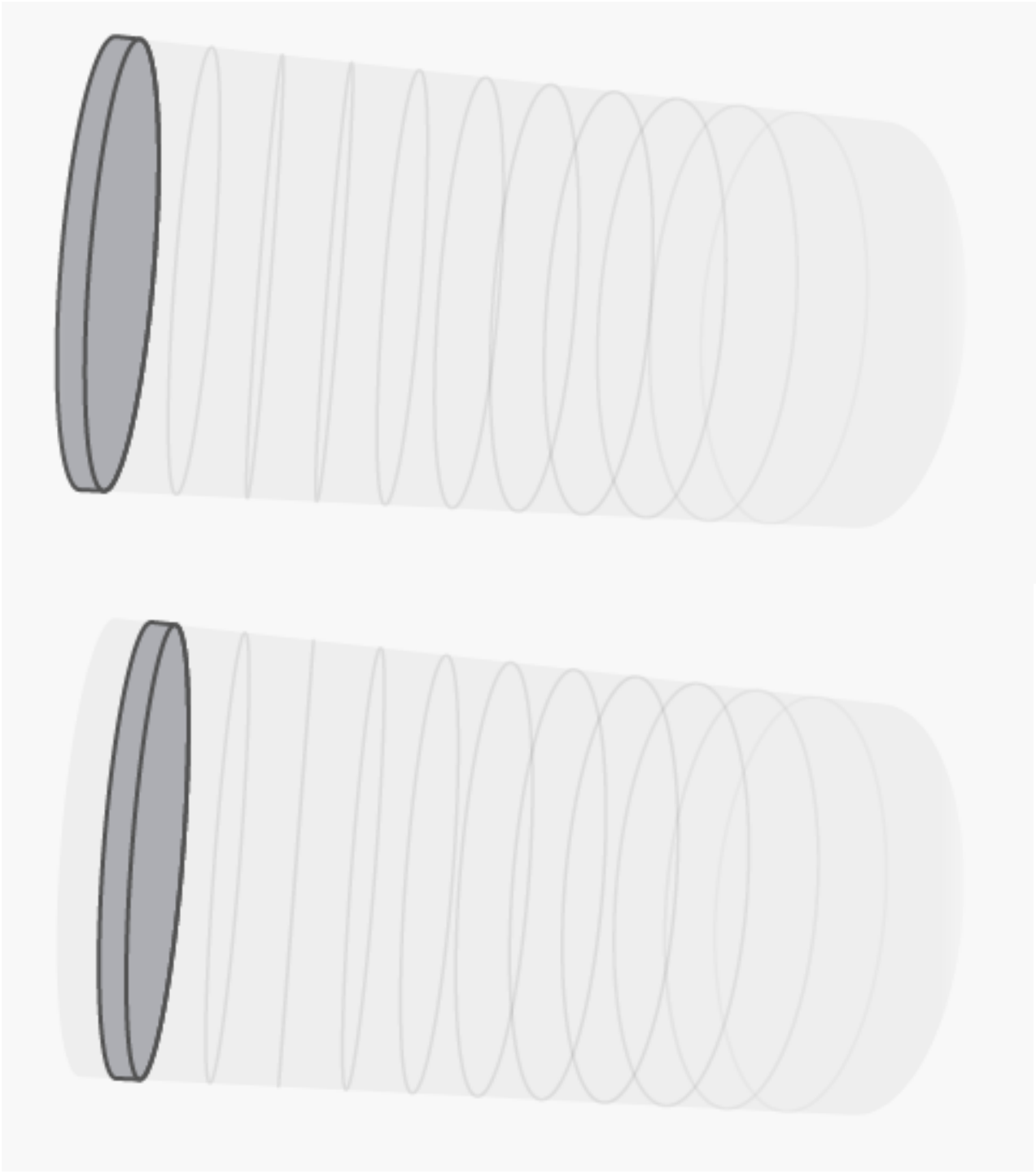
Sound is like a spirit that travels through our surroundings and carries beautiful music or noise sometime. The medium which is commonly associated with sound is Air. It comprises of minuscle particles (primarily Nitrogen and Oxygen). They move with the average velocity of 460 m/s. Those particles collide with each other or other things like wall of a given container. When a particle collide with wall, it imparts a small force. A large-large-large count of particles can cause the wall to expand, imparting enough force on it. That total force is simply a division of sum of all forces by area of the given container.

Latency due to Collision

When the container is resized to be smaller, the number of collisions apparently increases. Molecules move slowly due to collisions, making it take time for particles to travel far. Think of when someone sprays perfume in a room – it doesn't instantly reach us because aromatic particles need to collide with the air around our nose.

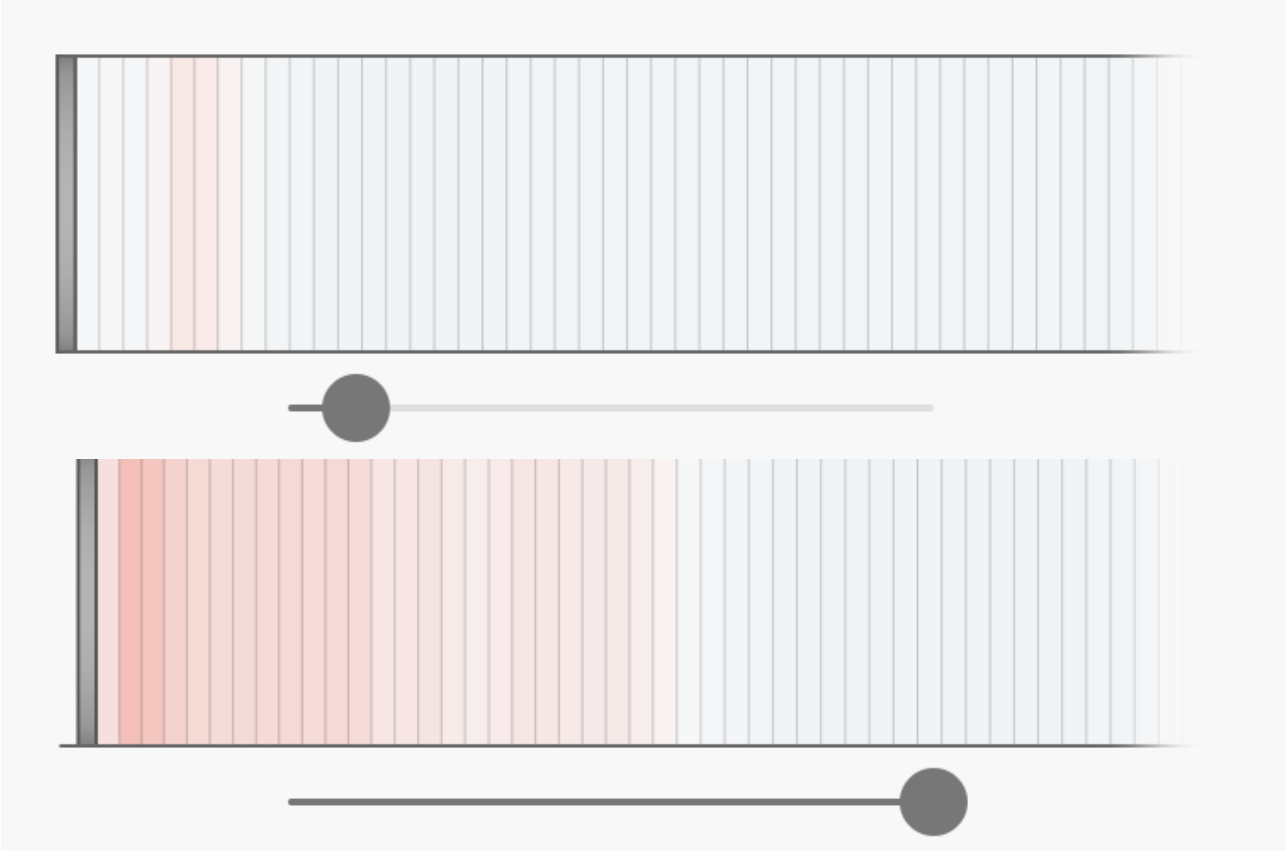
Speaker Intuition

Despite air particles moving fast, collisions with others keep them mostly in their local area. While over time they mix, in short timeframes, each air parcel contains mostly the same particles. The collective behavior of individual particles contributes to air properties at larger scales. Watch how air reacts on a larger scale when a moving plate pushes on a section of air in a tube. Imagine dividing the air near the circular plate into slices, represented by thin lines. Despite the slices being imaginary, they help us see what occurs in different air sections as the plate moves.



Here, individual air parcels move slightly from their starting position, but the disturbance caused by the plate's motion seems to spread through them at a certain speed.

Importantly, this disturbance moves away from the plate regardless of its direction. Let's look at this situation from the side, which will make it



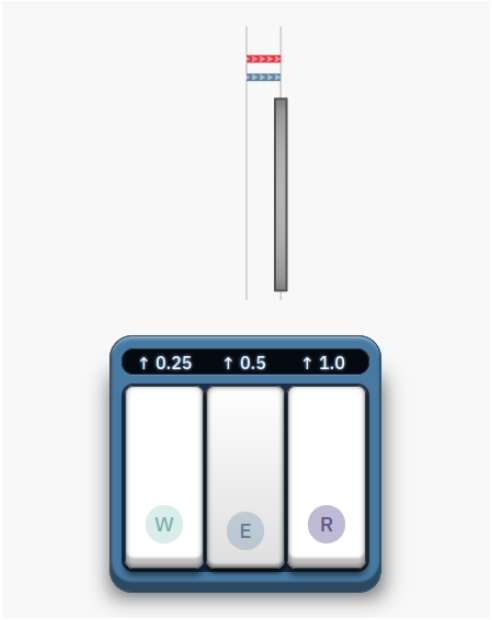
easier to see how these pressure changes propagate. When the plate moves to the right, it compresses nearby air particles, raising the pressure in that area. These compressed particles then push into

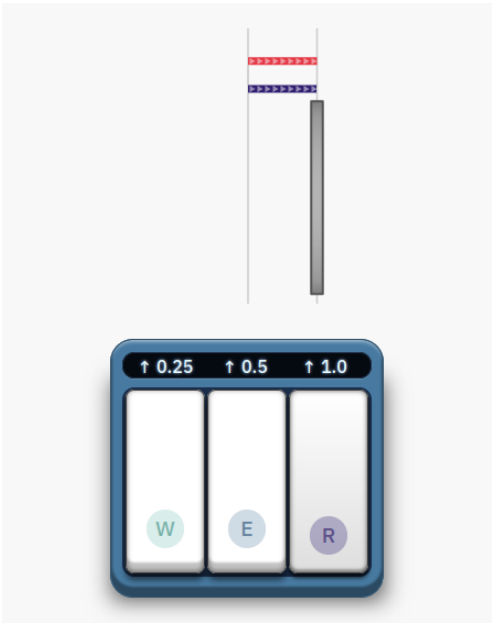
the adjacent slice of air, causing a chain reaction that results in a higher pressure zone moving away from the plate. Conversely, when the plate moves left, it creates an area of rarefaction, and the neighboring air rushes to fill the void, reducing pressure in that region and beyond. Although the slices move toward the receding plate, the pressure disturbance still moves away from it. This process we observed is essentially pressure waves traveling through the air, and when these waves reach our ears, we perceive them as sound. By moving the plate, we've created a basic speaker, emitting sounds as its moving part changes position.

When something shakes or moves (vibrates), it can make the air around it push and pull (pressure disturbances). Let's just think about how the movement of the shaking thing is connected to the sound it creates.

It sounds good

Let's talk about a simple music-making device. When we press a key, it makes a plate quickly move, and when we release the key, the plate goes back. Each key makes the plate move by a different amount, shown by a red marker.





REFER : https://ciechanow.ski/sound

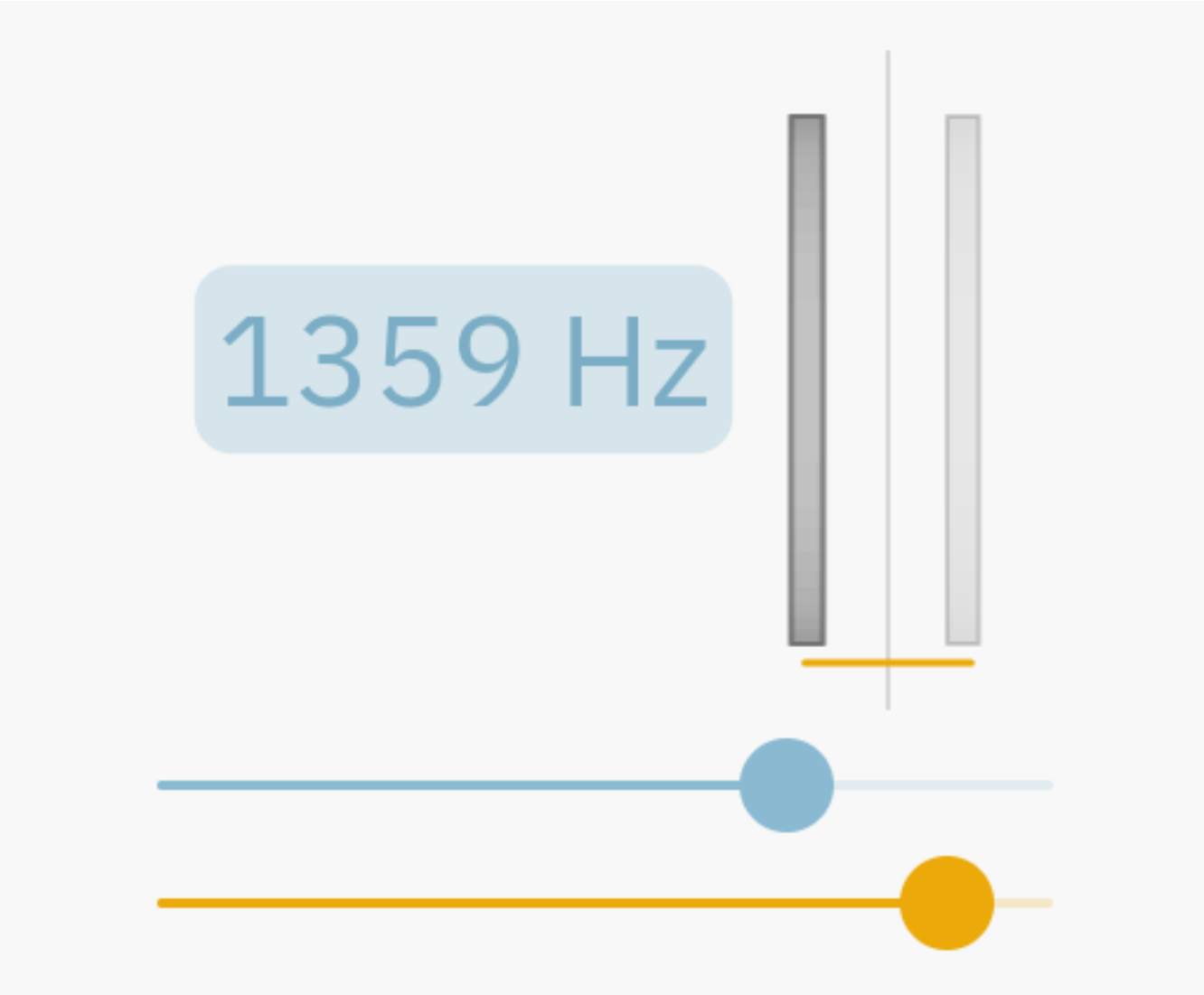
This simple device teaches us a few things. First, a sudden plate movement creates a pop sound, and

it sounds the same whether the plate jumps forward or backward. Second, the pop's loudness depends on how much the plate moves – the bigger the jump, the louder the pop. We can relate this to the thunder sound by lightning which travels by tearing the air.

However, making music with just these pops has limits because it only allows for simple beats, and pressing buttons too fast is limited by how fast fingers can move.

The Jargons

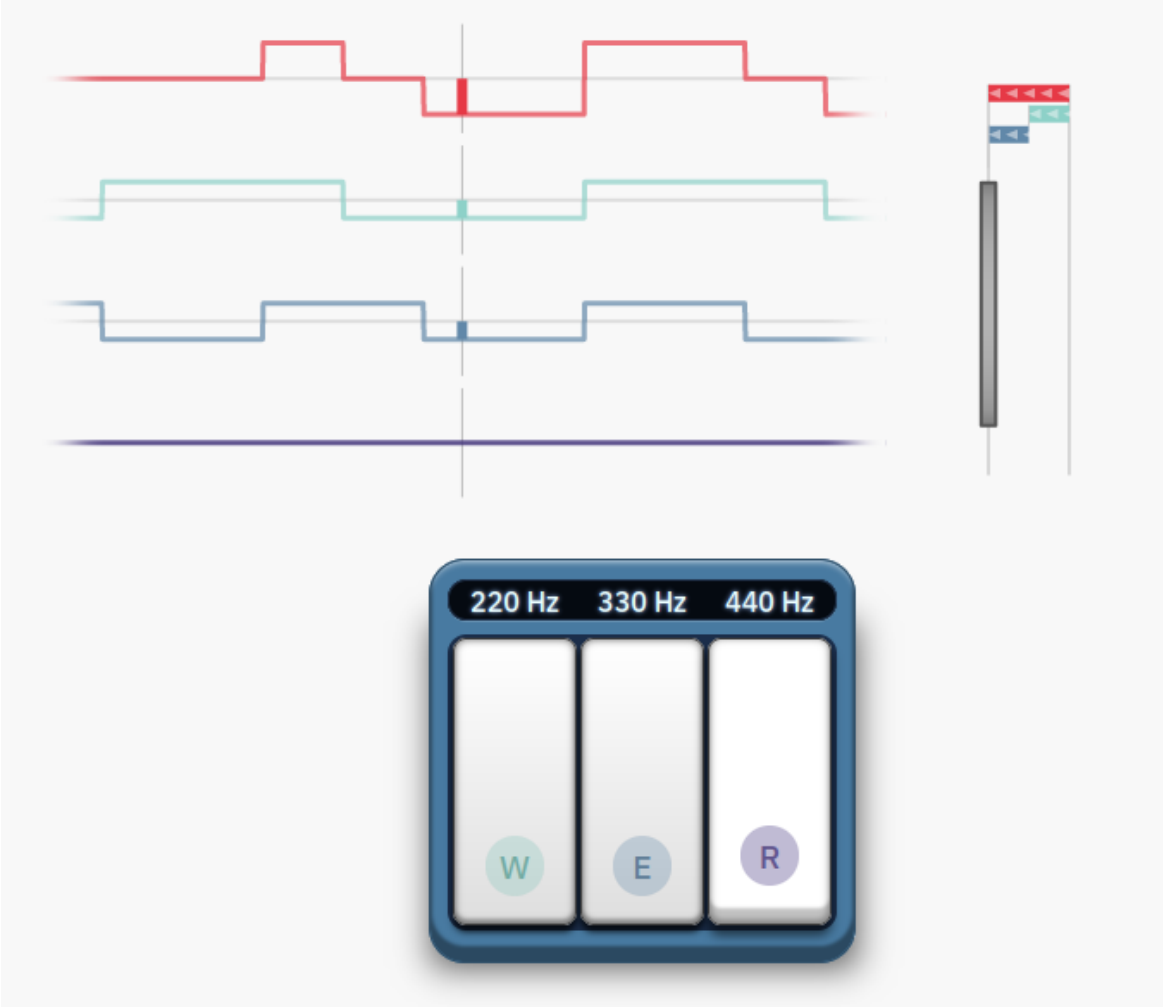
If we can ask the device we're using to repeat the pops at a certain rate. In the demonstration below, we can drag the first slider to change how many times per second the plate should jump forwards and backwards.



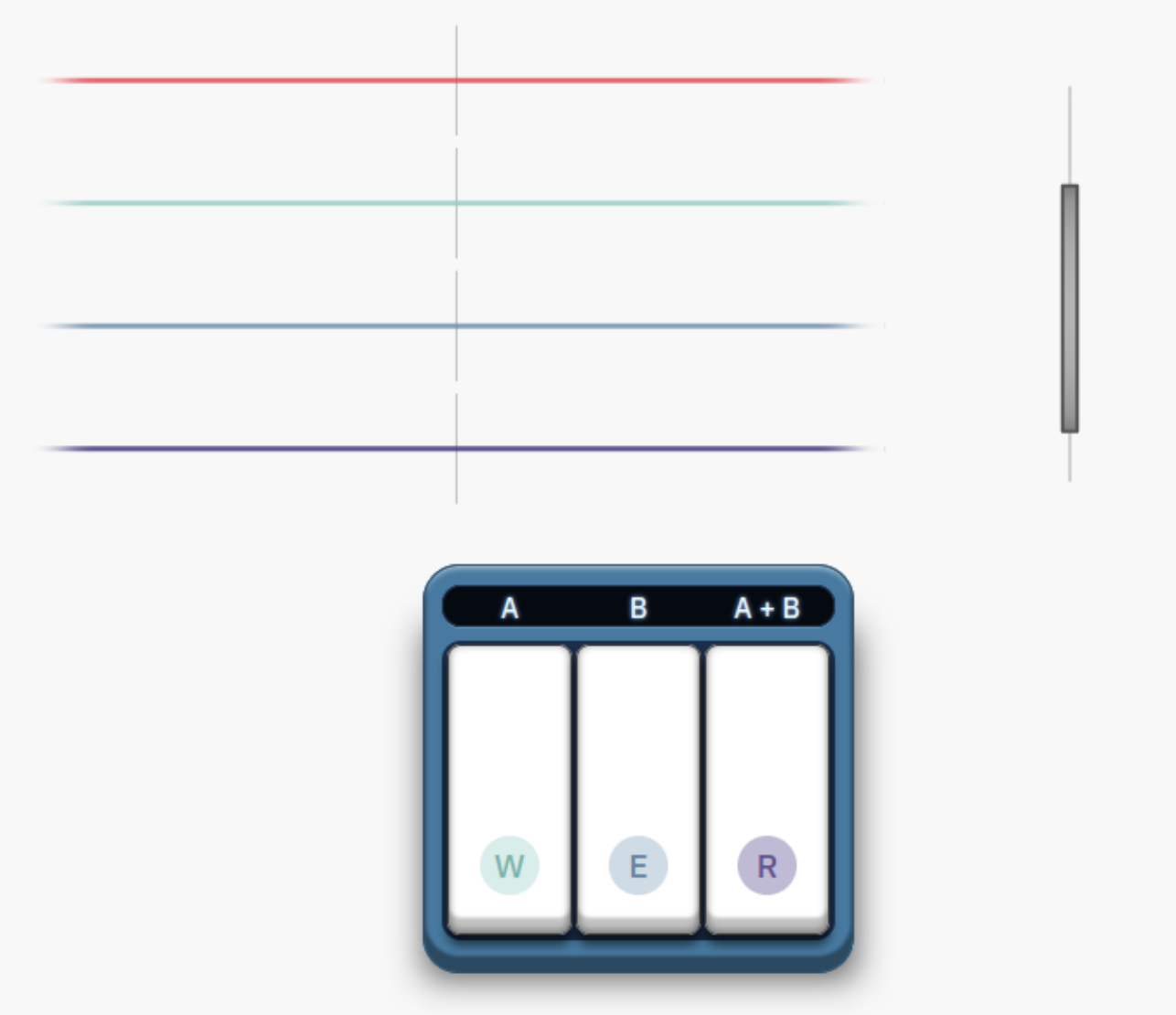
REFER : https://ciechanow.ski/sound
To talk about how fast something shakes, we use hertz (Hz), which tells us how many times it moves back and forth in one second. For example, if it has a frequency of 2 Hz, it means it moved back and forth twice in a second. At lower frequencies, the movement sounds like quick pops, but at higher frequencies, the sounds blend into a note. The key point is, the higher the frequency, the higher the pitch of the sound. Pitch is how high or low a sound is. When something makes a high-pitched sound, it's like a squeak or a chirp. A low-pitched sound is deeper, like a rumble or a bass drum.

Now, let's make our music device create frequent jumps of the plate. Each key has a different frequency, shown above the keys with three plots for each key, and an extra one showing the combined effect of all the movements. These plots show how the plate moves as each key is pressed.

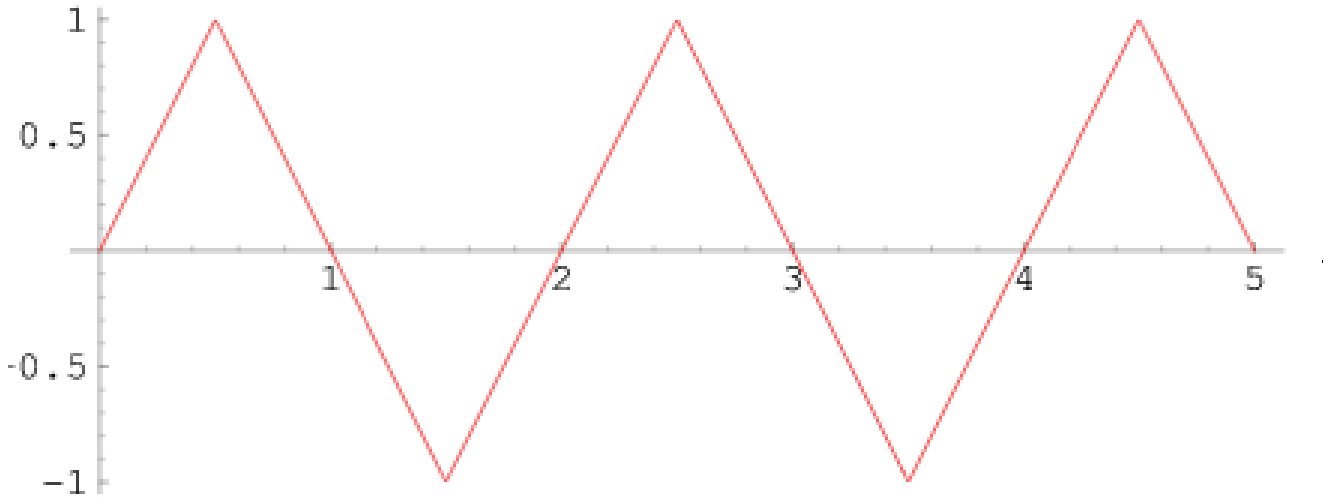
Now, when we press two keys simultaneously, we can hear both sounds because the final movement of the plate is the combination or sum of what each key is asking it to do. If one key wants the plate to move up while the other wants it to move down, the actual movement is the result of adding these requests together.



This combination of movements creates the overall motion of the plate that we hear as the combined sound when multiple keys are pressed. Here, we have the third key which makes sound in the frequency sum of A and B.



The sound emitted when the third key is pressed is exactly the same as the sound emitted when the first two keys are pressed together. This is a very important observation, as it lets us explore some other patterns of motion that the keys could induce. Let's change how the plate moves by making it go at a steady speed in a triangular pattern instead of rapidly jumping back and forth.



This pattern, like other shapes that repeat, has three main characteristics: amplitude, frequency, and phase.

Amplitude : how tall the pattern is.

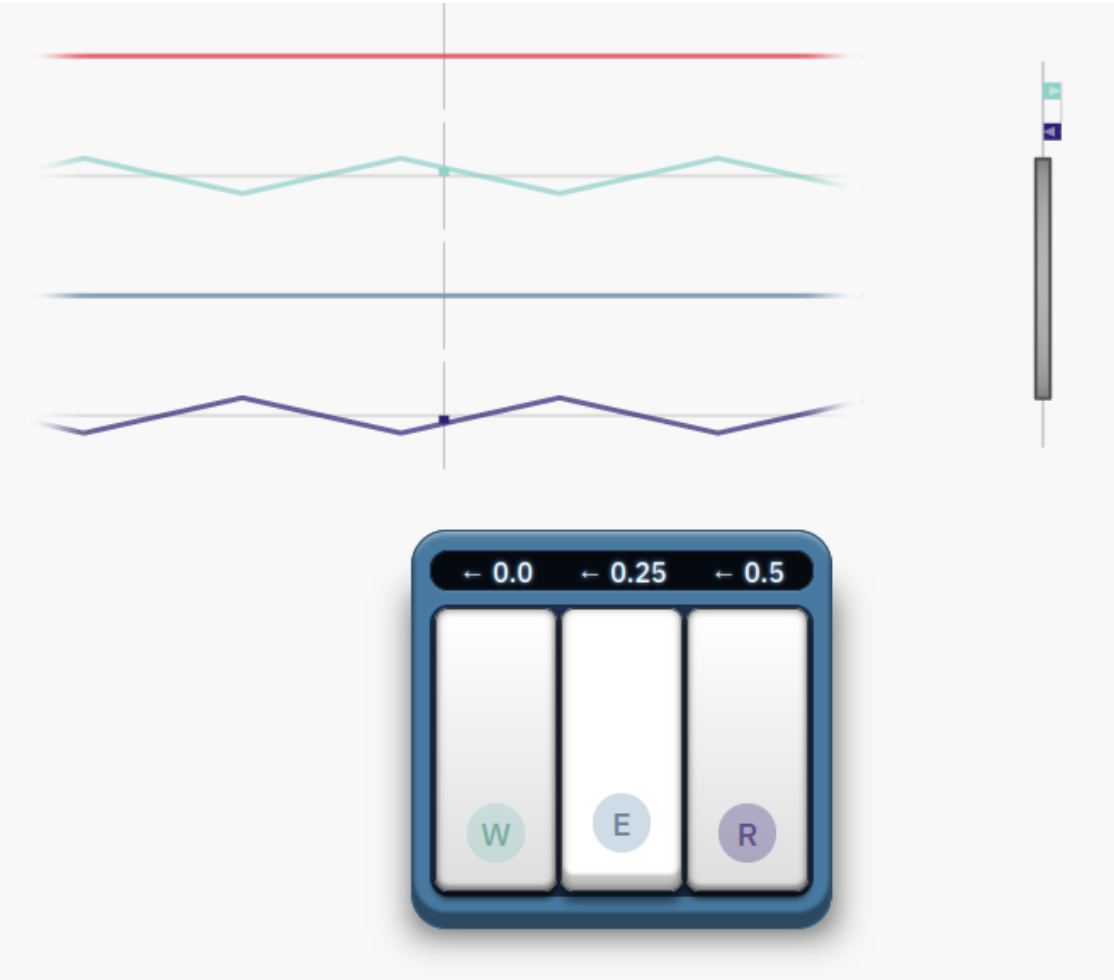
Frequency : how wide the repeating shape is, deciding how many times it goes up and down in a second.

Phase: just shifts the pattern around, changing where it starts.

We know that the amplitude (height) affects how loud the plate sounds, and this triangular motion is the same. Similarly, The higher the frequency the higher the perceived pitch.

Destructive Interference

Imagine we have three keys on a musical instrument as shown below, and when we press each key by itself, they make the same sound. No surprises there. However, when we press two keys together, things get interesting. If we press the middle key with either the left or right key, the sound gets a bit louder. That makes sense.

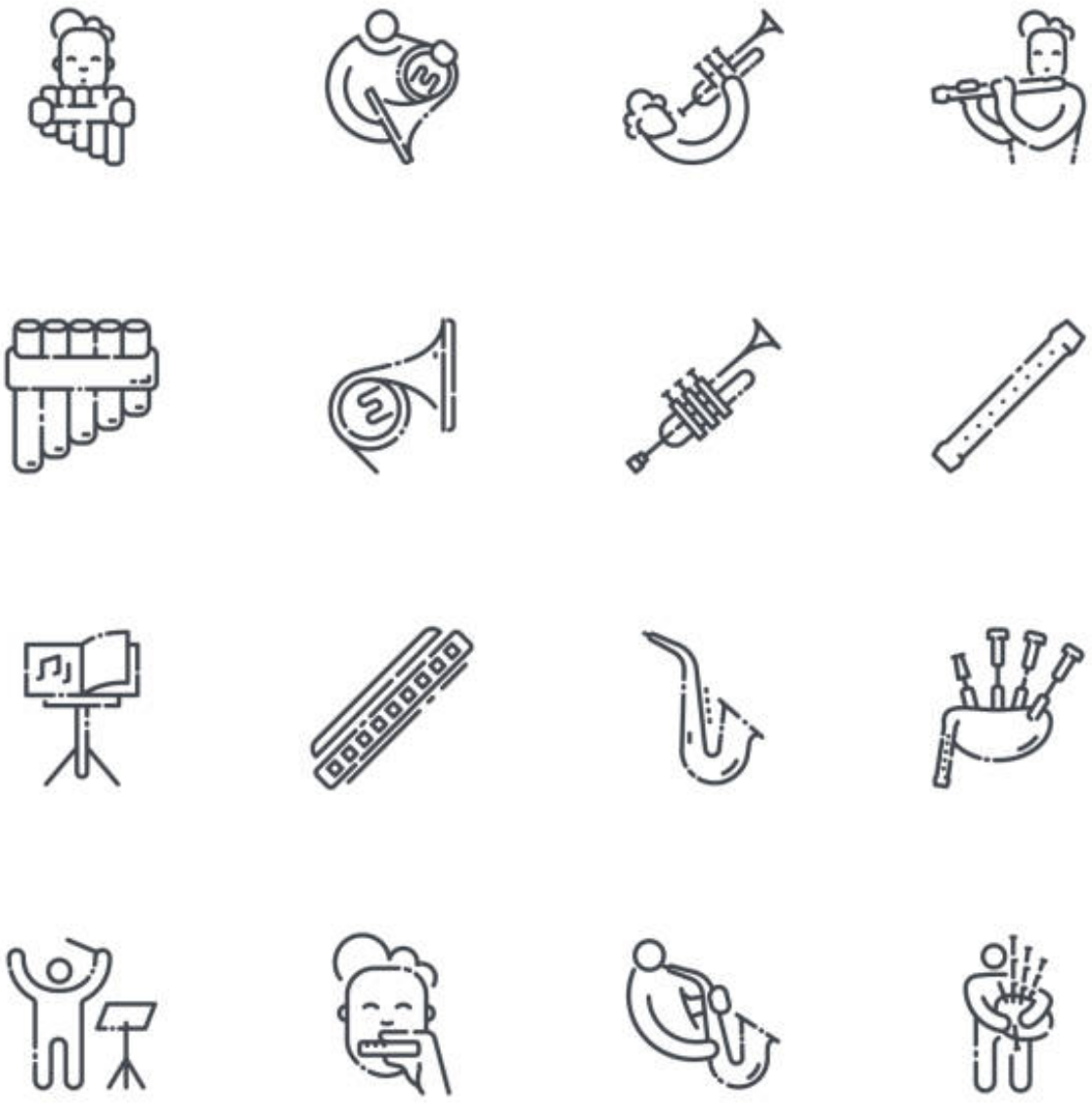


But, when we press W and R keys at the same time, there will be no sound produced. This is due to destructive interference. Destructive interference occurs when two or more waves meet and their amplitudes combine in a way that decreases the overall amplitude of the resulting wave. In the context of waves, including electromagnetic waves, sound waves, or water waves, destructive interference occurs when the peaks of one wave align with the troughs of another wave.

The Timbre

Timbre, also known as tone color or tone quality, is a complex attribute of sound that distinguishes different types of sound production. It is what allows us to differentiate between two sounds even if they have the same pitch and loudness. Timbre is often described as the color of a sound and is a subjective quality that contributes to the richness and character of a musical note or sound. Different musical instruments or sound sources produce unique timbres. For instance, a trumpet sounds different from a flute, even when playing the same pitch and volume.

Timbre is what allows us to differentiate between different instruments playing the same note. The reason different waveforms have different timbres lies in their harmonic content.



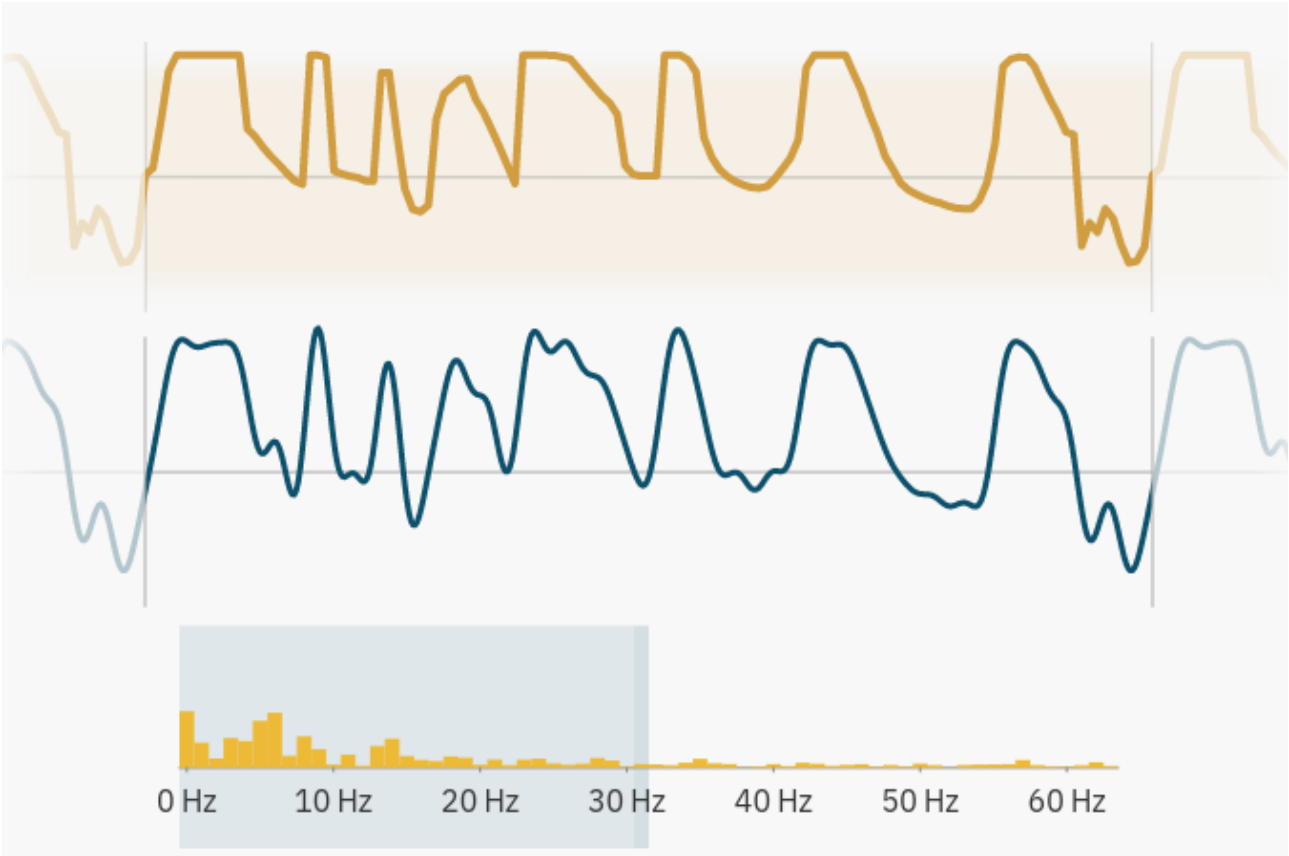
Pure Tones

Sinusoidal waves, often referred to as sine waves, are fundamental periodic waveforms that play a crucial role in the study of wave mechanics and signal processing. A sinusoidal wave has a smooth, repetitive oscillation resembling the shape of a sine function. The waveform is characterized by its amplitude, frequency, and phase. The general form is $y(t)=A\cdot\sin(2\pi ft+\phi)$, where:

- A is the amplitude,
- f is the frequency,
- t is time,
- ϕ is the phase.

When we press two keys of sound generator that makes sinusoidal waves, each with minute difference in frequency, we'll be able to experience beats.

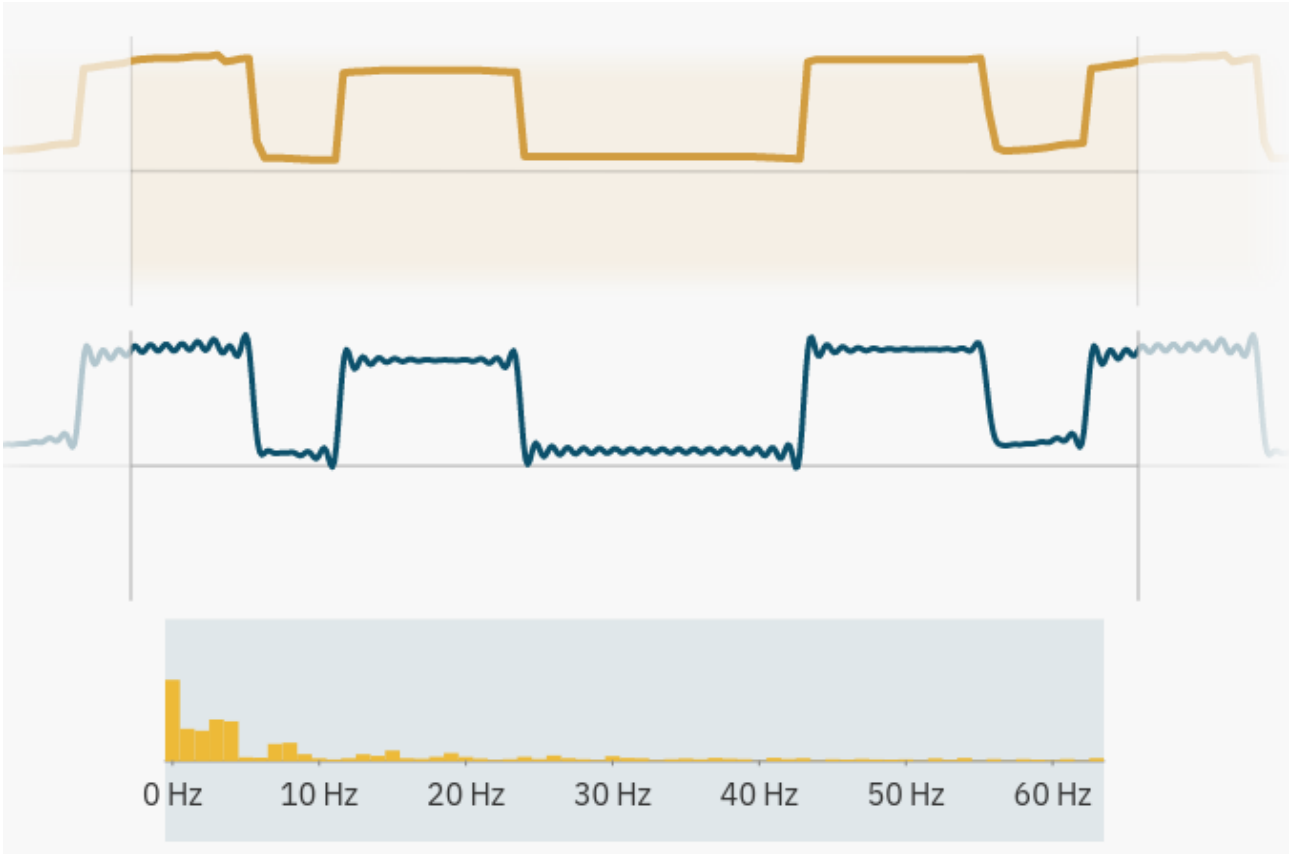
Surprisingly, we can break down any repeating waveform into its constituent sine waves. The bottom part shows the amplitudes of individual sine waves creating the final shape.



This method is known as Fourier series or Fourier transform, allowing us to deconstruct a waveform

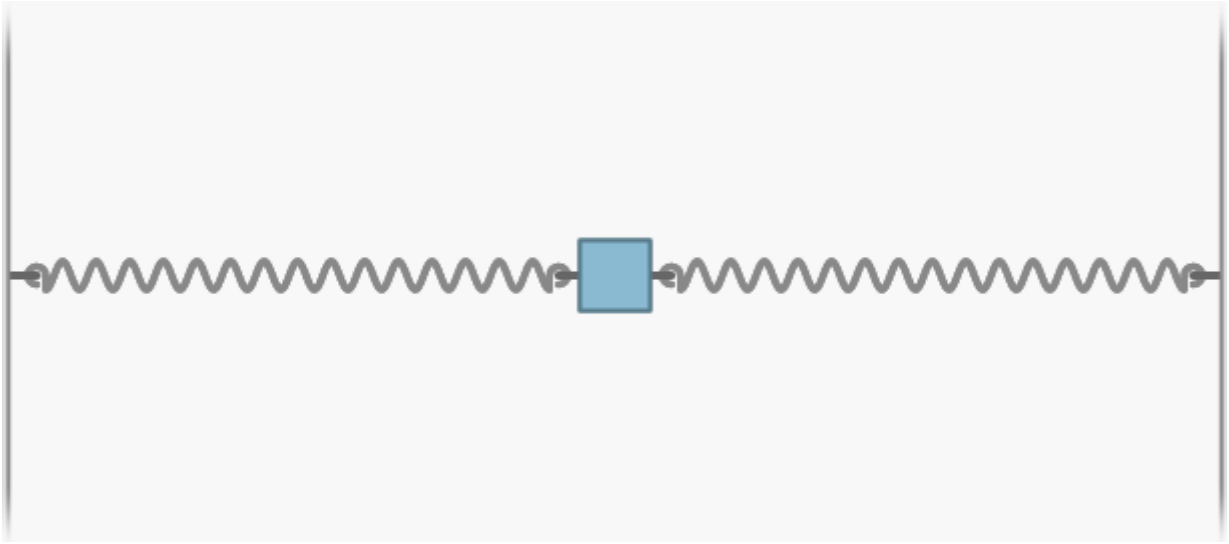
expressed over time into individual frequencies contained in that waveform. It provides two views: one as a collection of values over time and the other as a collection of distinct sine waves adding up to the same pattern. Fourier transform gives both amplitude and phase of each contributing sine wave. However, as human hearing is less sensitive to phase shifts, amplitudes typically suffice to express the original shape. Gibbs' phenomenon occurs near a jump discontinuity in the signal. It says that no matter how many terms you include in your Fourier series there will always be an error in the

form of an overshoot near the discontinuity. The overshoot always be about 9% of the size of the jump.

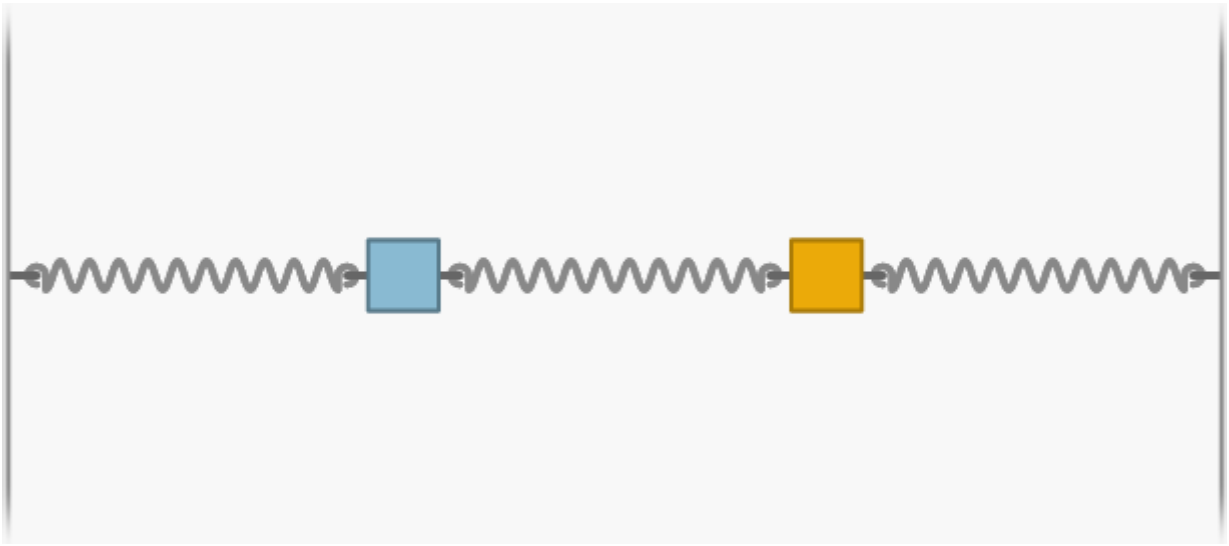


Masses and Springs

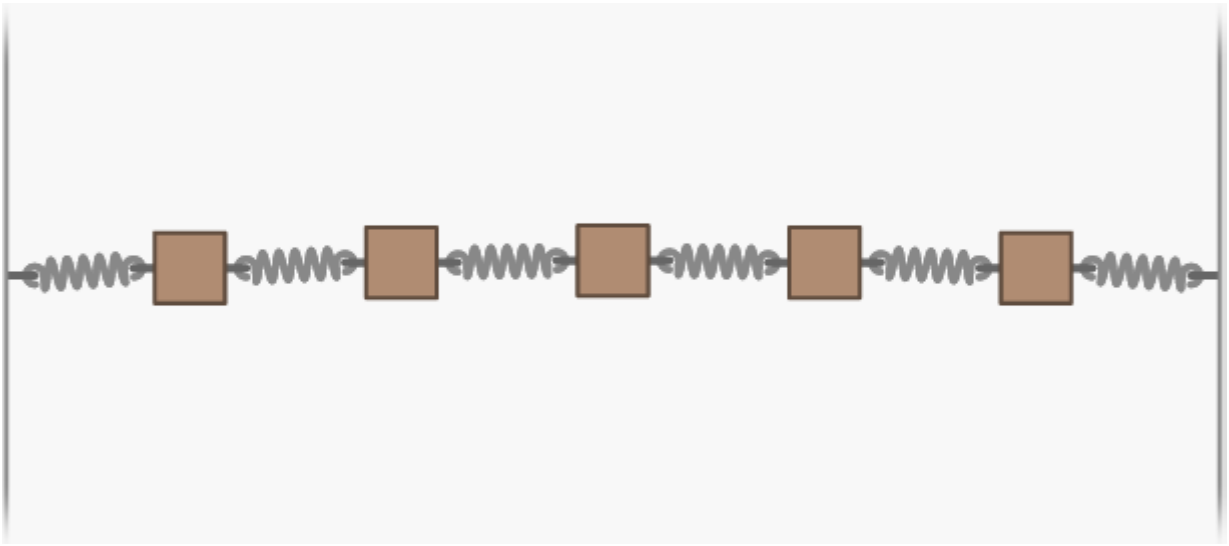
Consider the following system. At small amplitudes, the system oscillates sinusoidally with a natural frequency determined by the mass and spring stiffness. Friction and air resistance cause the oscillations to diminish over time.



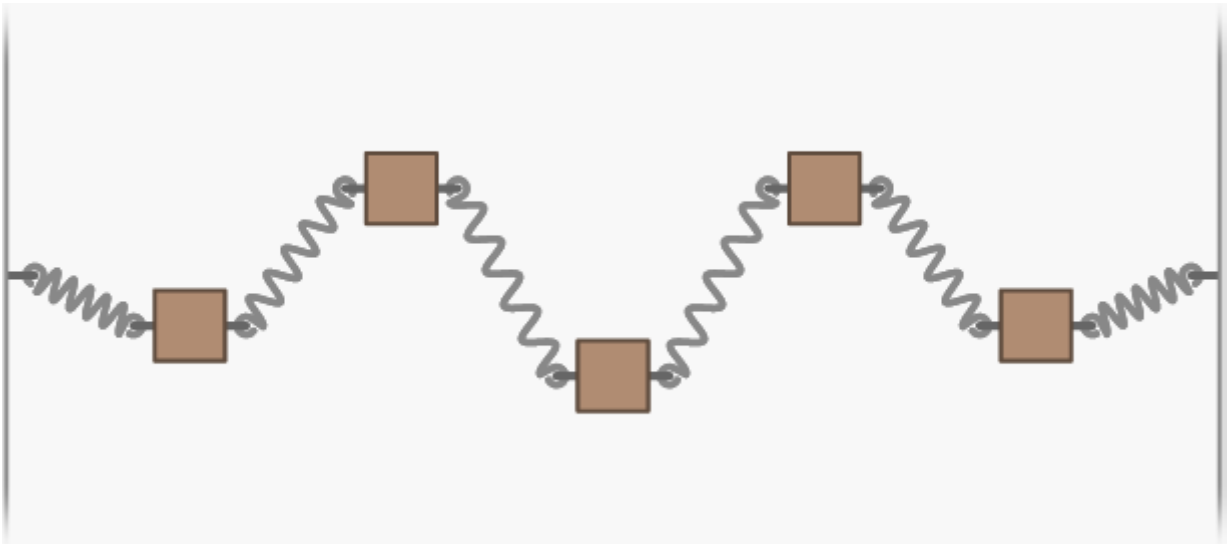
Adding another mass and spring complicates the system, resulting in intricate motion patterns.



The masses either move together or in opposite directions. This method extends to more complex systems. For a system with five masses and six springs, we can observe the five normal modes in the demonstration. As we increase the number of masses and springs, the system resembles tiny sections of a string.

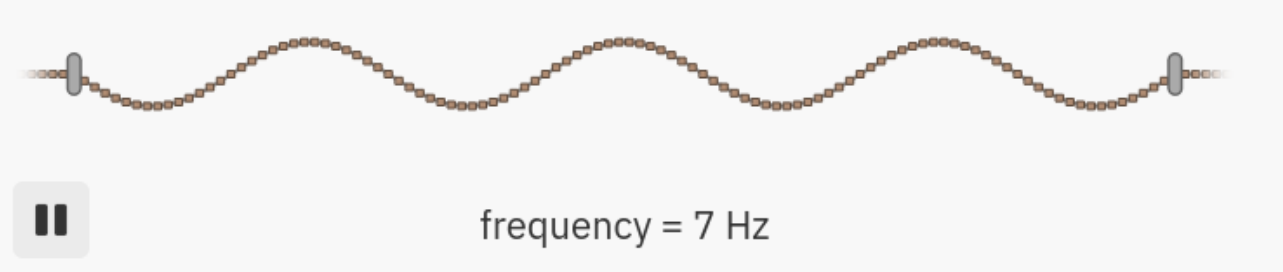


Considering a simple string as a system of many small spring-and-mass segments, the first few normal modes of this system are demonstrated. Only certain wave patterns are allowed due to fixed endpoints, and these modes of vibration are known as Harmonics.



Similar to Fourier transform operations for waveform shapes, the motion of a vibrating string can be expressed as the sum of individual harmonic modes, each with a different amplitude and a frequency multiple of the fundamental frequency. A demonstration allows us to pluck the string, and you'll notice changes in sound due to different harmonic ratios caused by the plucked string's shape. Placing a

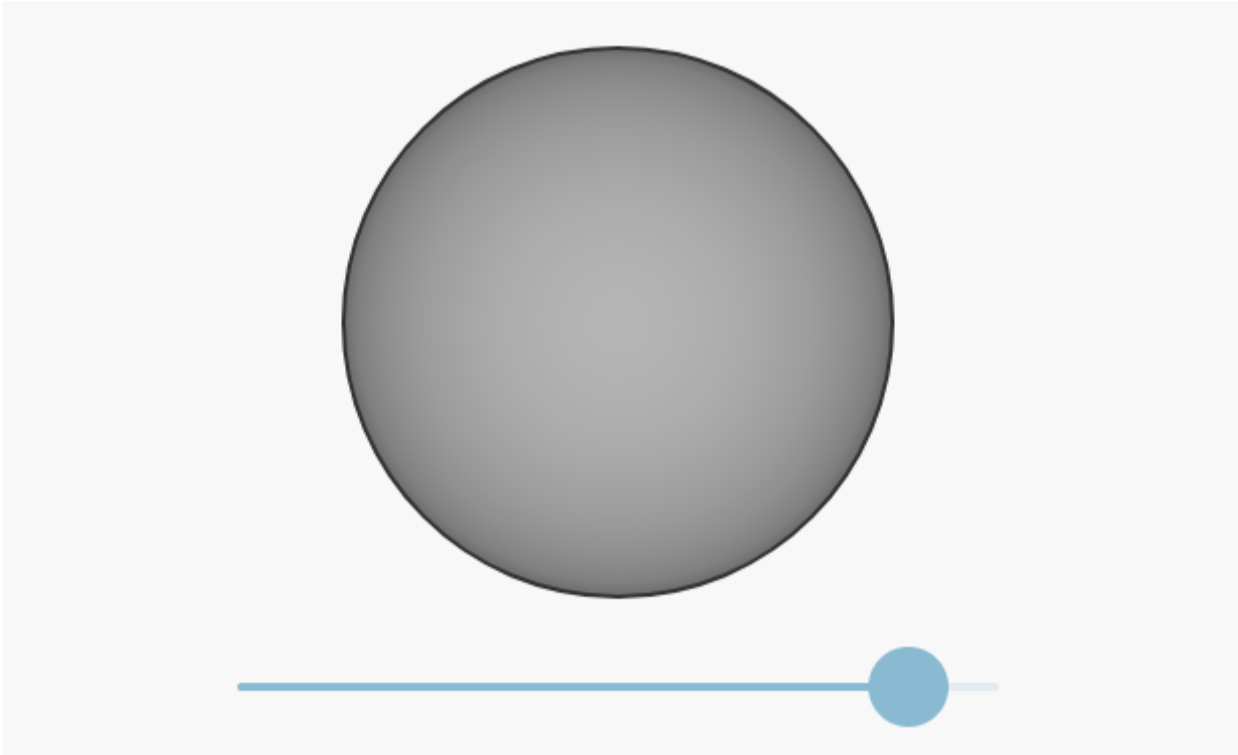
finger on the string alters its effective length, producing different notes.



The thicker and heavier the string, the lower the frequency. We can explore this in a demonstration where we change the string's thickness or material.

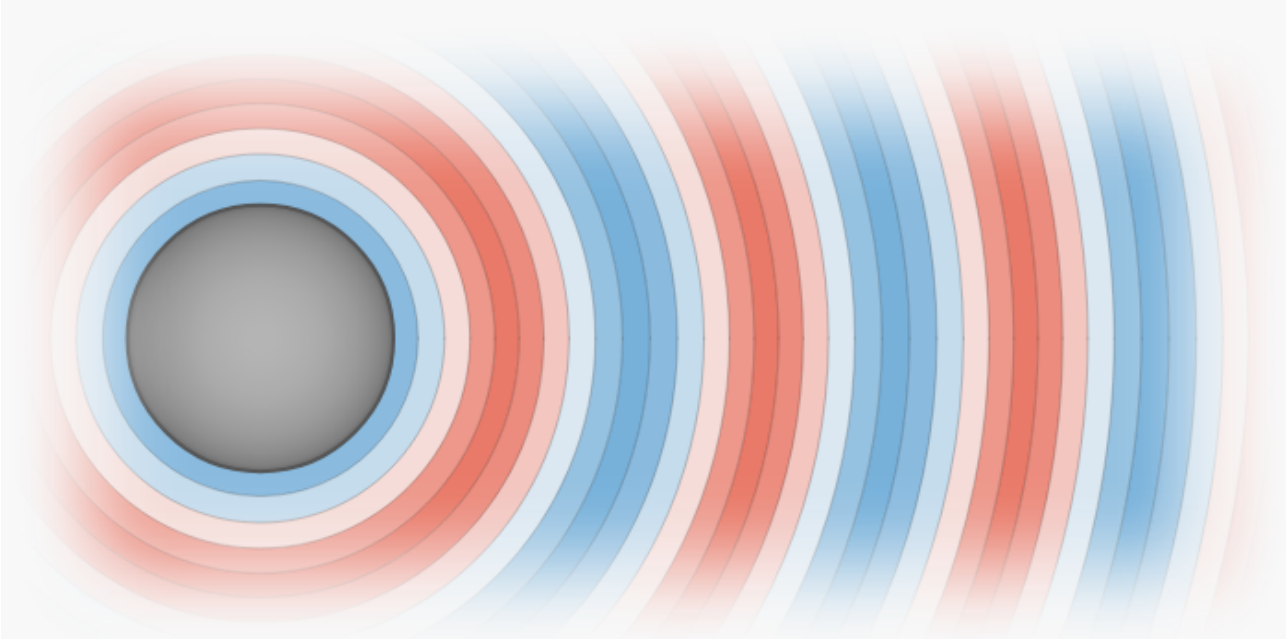
Pressure Waves

Imagine a water-filled balloon that we can quickly fill or empty. In the demonstration below, we can observe a pulsating sphere controlled by a slider that adjusts its vibration frequency.



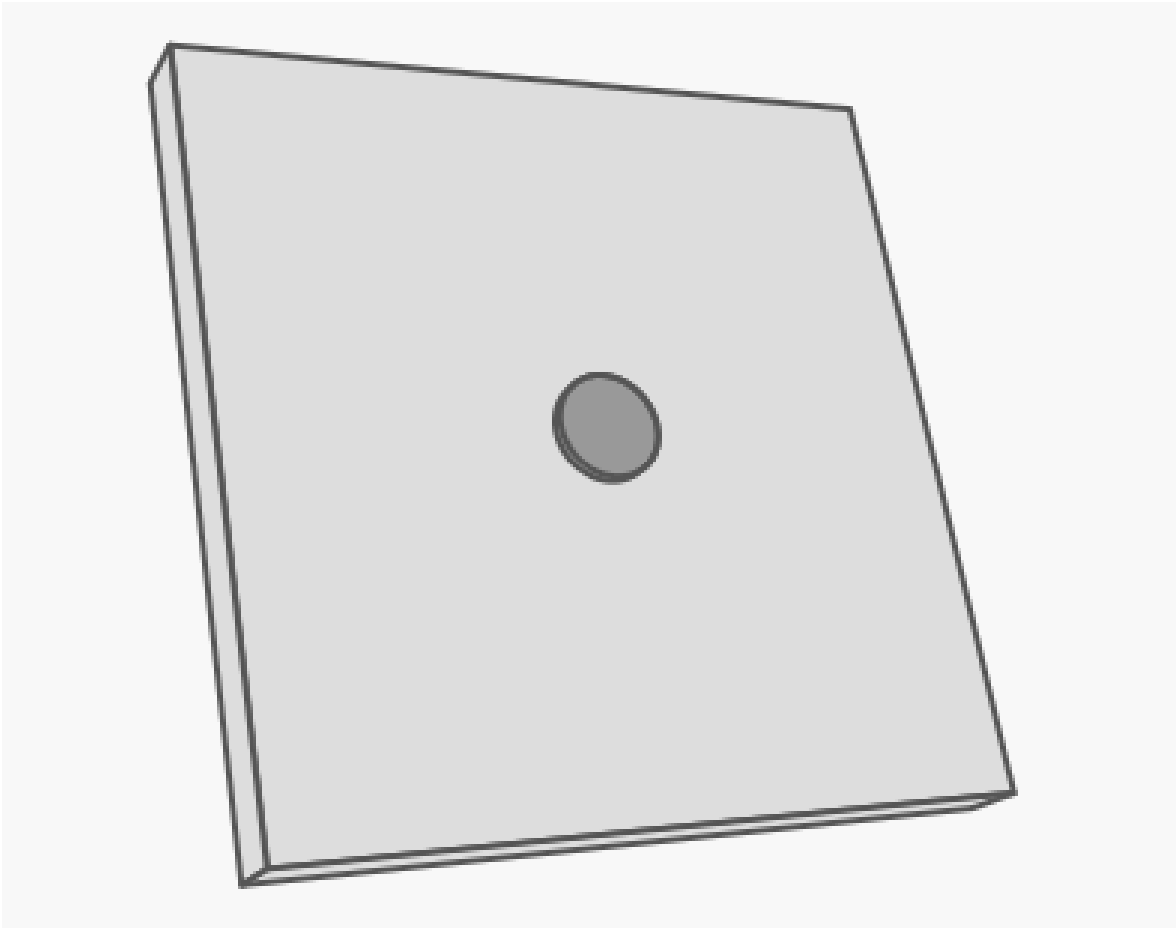
To understand the pressure near the surface of this sphere, we'll use the method of slicing the air into thin

shells around the sphere. At high vibration frequencies, the air around the sphere behaves similarly to a simple plate in a tube, demonstrated below.

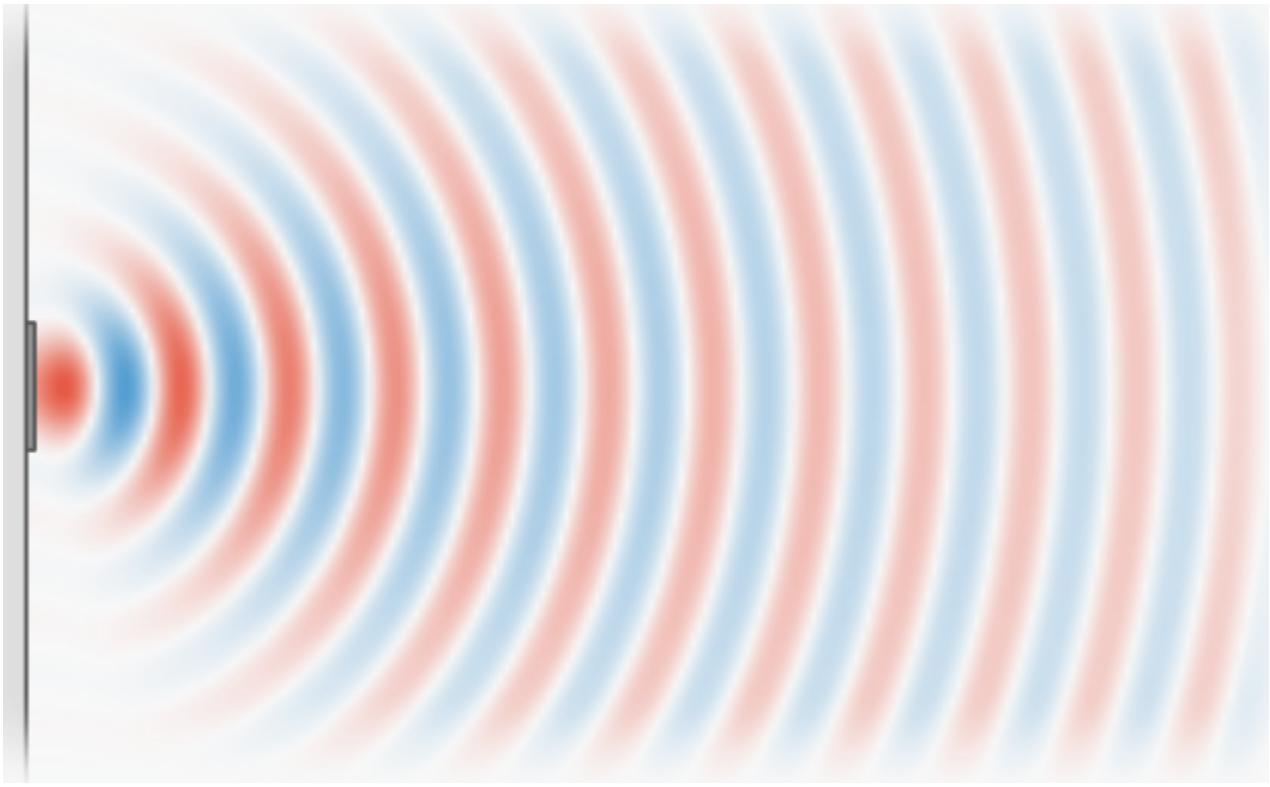


What we see here is the inertia of the air. When the sphere initially expands, it accelerates the nearby air. Even when the sphere stops growing, the air, still having some velocity, moves away, causing a local

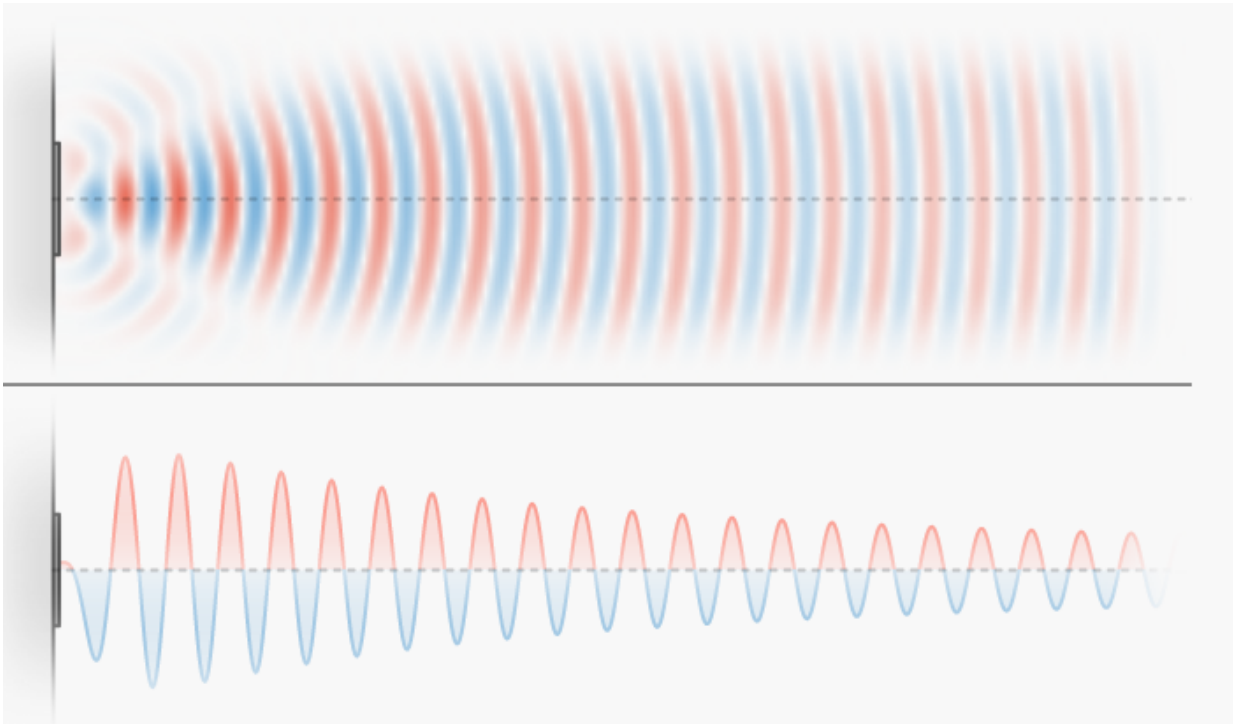
rarefaction. Likewise, as the sphere shrinks, the surrounding air is pulled towards it, continuing to bunch up even after the sphere reaches its smallest size. A sphere forms only a simple model of a speaker – a more realistic example would be a vibrating circular plate surrounded by a very large wall. Similar to a sphere, the connection between the



position of this plate and the resulting pressure also relies on the sound's frequency. However, while the sphere was perfectly symmetrical, a vibrating plate loses some of those symmetries. As a result, the intensity of the pressure disturbances now varies depending on the direction relative to the source. When an object vibrates, it creates pressure variations in the



surrounding air. The distribution of these pressure changes can be intricate, but at a moderate distance from the vibrating source, the oscillation of the object produces a regular pattern of pressure changes. Specifically, if the source is undergoing a steady up-and-down motion, it leads to a sinusoidal variation in pressure that propagates through the air. In the provided simulation, the dotted line in the lower part represents the "side" view of this pressure variation.

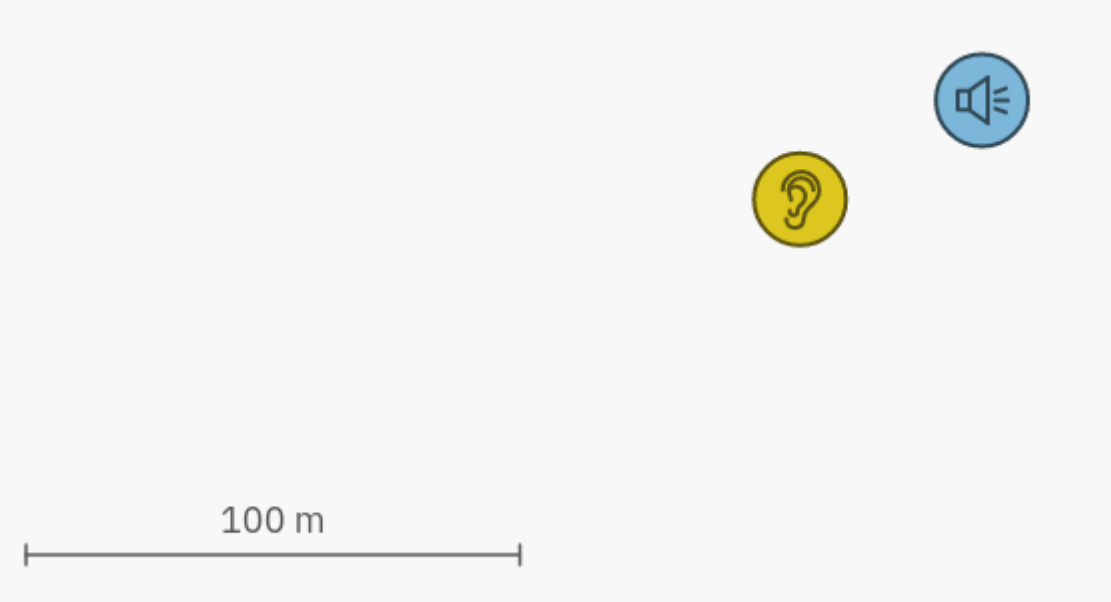


This means it shows how the pressure changes in the air as we move away from the vibrating source. The sinusoidal pattern indicates a regular and repeating oscillation in the air pressure caused by the continuous vibration of the object. If we're far from the source, we notice a delay, like when

lightning strikes. Also, how loud the sound is depends on how far we are. As we move away, the sound gets quieter. This happens because the waves that make sound spread out like circles. Now, when we talk about sound, we use something called Sound Pressure Level (SPL), measured in decibels (dB). It helps us compare sounds without using big numbers. The SPL equation looks like this: $SPL = 20 \times \log_{10}(p / p_0)$, where p is the pressure we measure, and p_0 is a reference value. If we double the distance from a sound source, the SPL goes down by 6 decibels. If we use two speakers

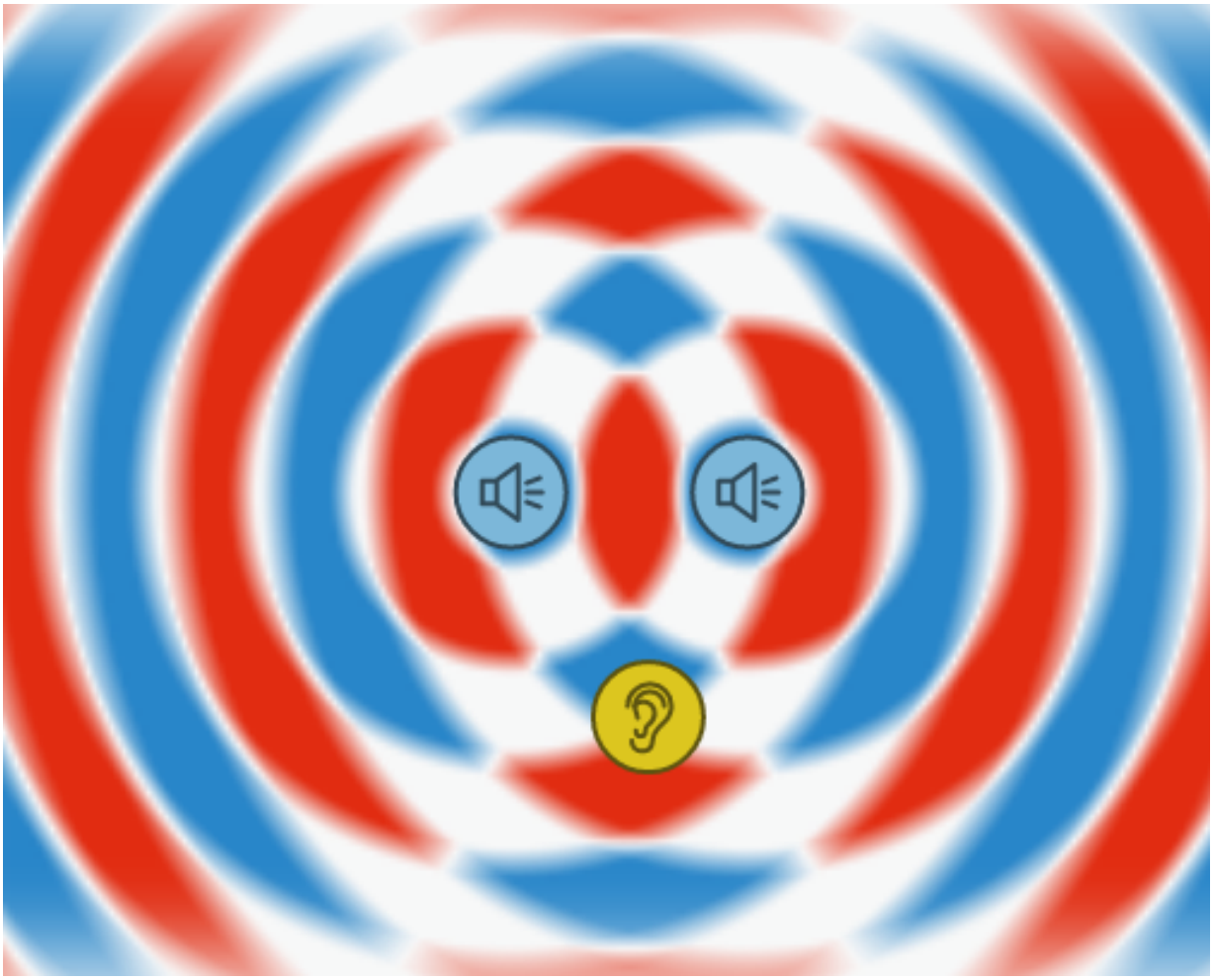
instead of one, the SPL goes up by 6 decibels. The loudness of a sound also depends on its frequency, like when we played with the moving plate. So, we sometimes weigh the measurements to match how humans hear. There's another thing called sound intensity, which measures the power of sound per area. It follows the inverse square law: $I \sim 1 / r^2$. We can use this to calculate the Sound Intensity Level. The Doppler Effect is like a change in sound or light when something is moving. Imagine we're standing still, and a car with a honking horn is coming towards you and then passes by. As the car gets

closer, the sound of the horn is higher, and when it moves away, the sound gets lower. This change in pitch (high or low sound) is the Doppler Effect.

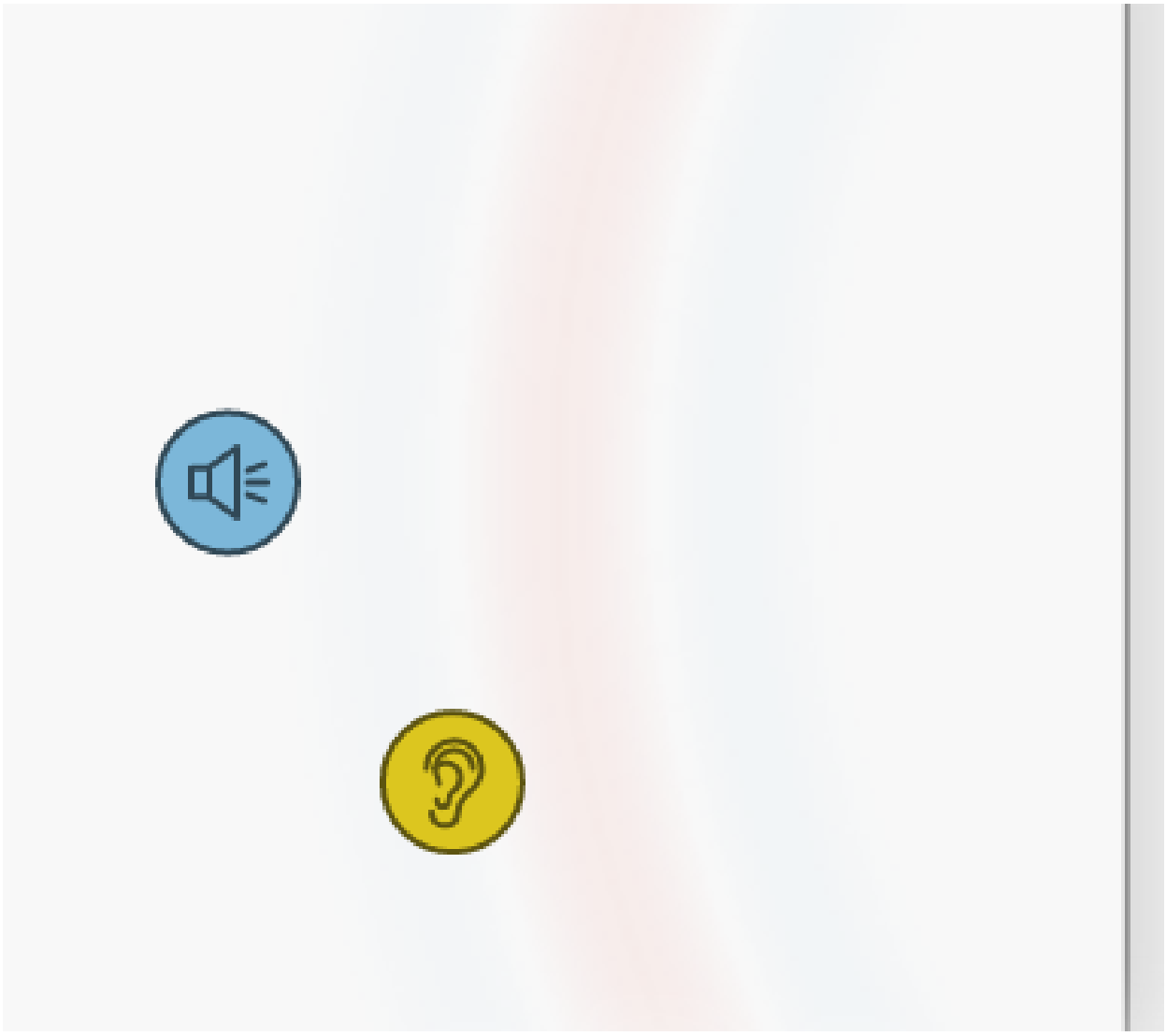


The distance affects how fast the sounds reach us.

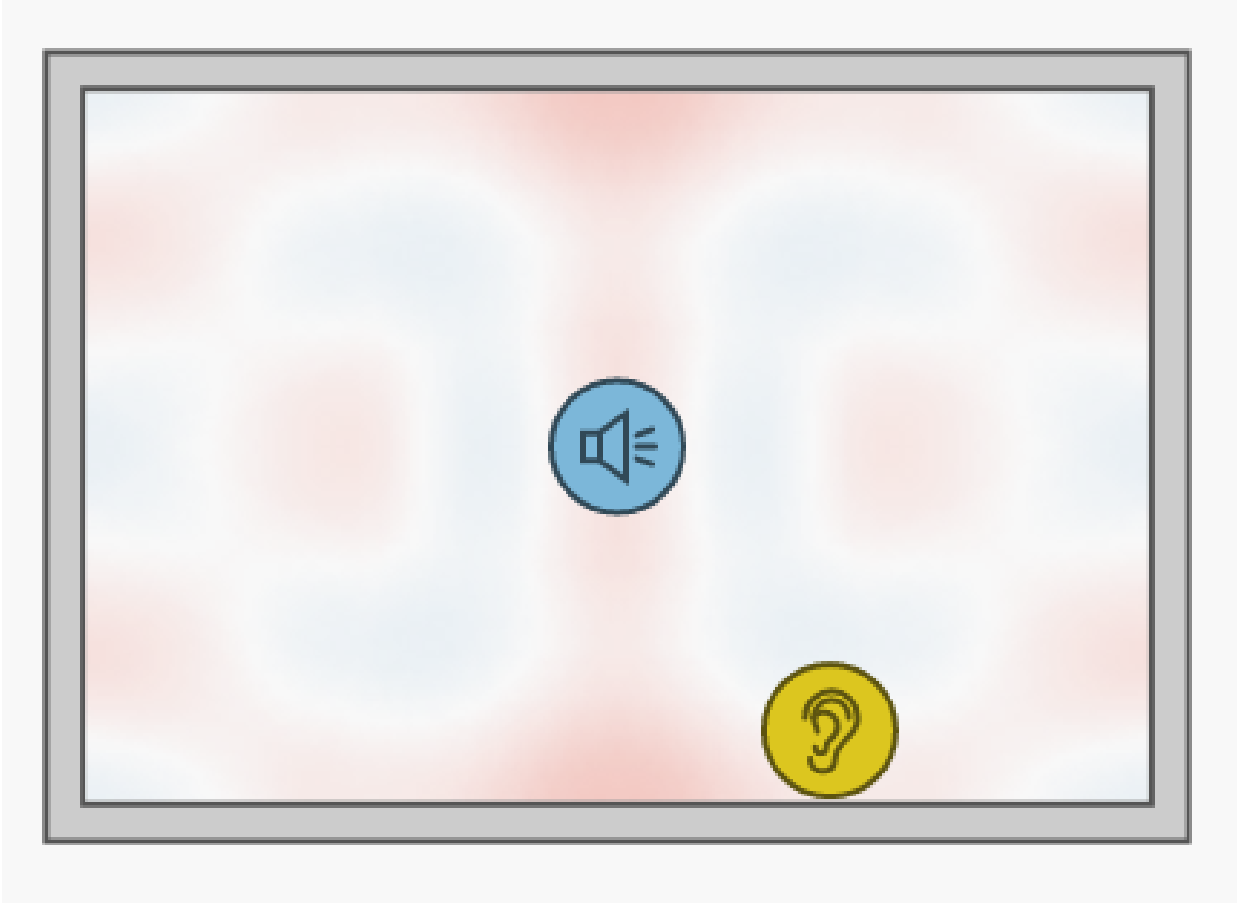
Even if things aren't moving, interesting things can still happen. For example, if two speakers play the same notes at the same time, the sound can get louder or quieter in different places. This is because the sound waves from one speaker can cancel out the sound waves from the other in some spots.



When there's a wall, the sound can bounce back, creating an echo. The closer we are to the wall, the shorter the delay between the original sound and the echoed sound. The wall also makes the echoed sound quieter because it absorbs some of the energy.

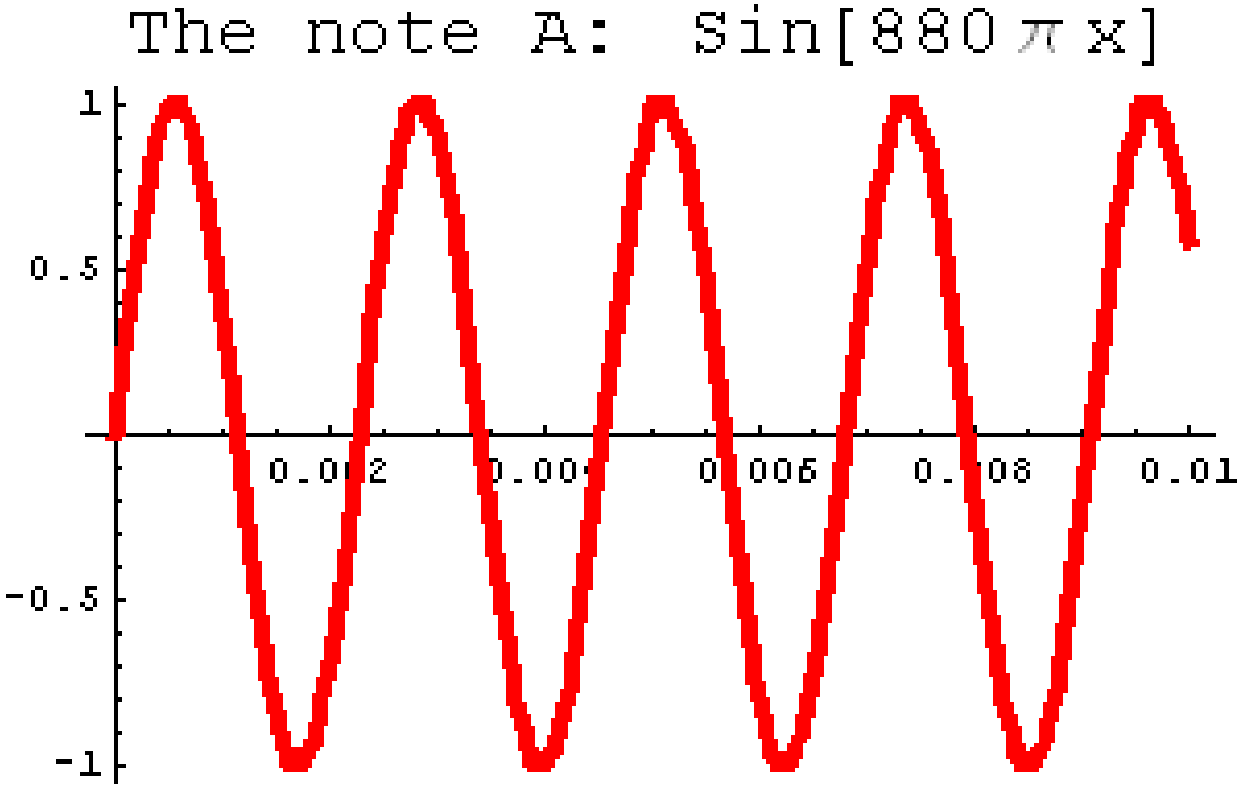


In a closed space, like a room, sound waves bounce off the walls and keep going for a while, creating an effect called reverb. Even after we stop making sounds, the echoes continue until they lose some energy. This is why we might hear reverb in an empty room.



Numbers into Audio

Sound to a computer can be thought of as a simple wave encoded digitally. Before the sound reaches your ears it goes through a digital to analog converter, essentially translating the digital signal to a current for your headphones / speakers.

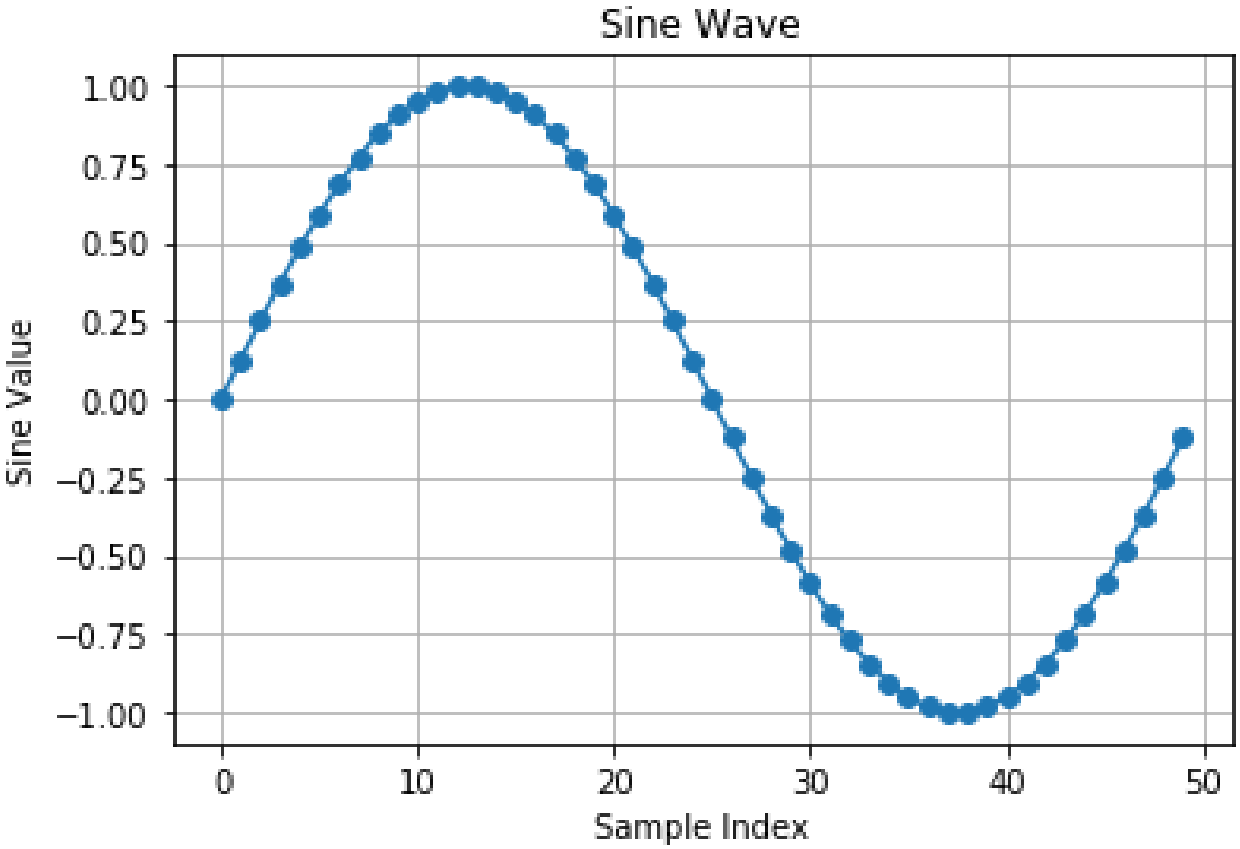


As a first step, let’s try to just create a sine wave with Python. We can generate this using math.sin(x) and pass x as radians. We’ll have to iterate over a range to get the sine wave out of this.

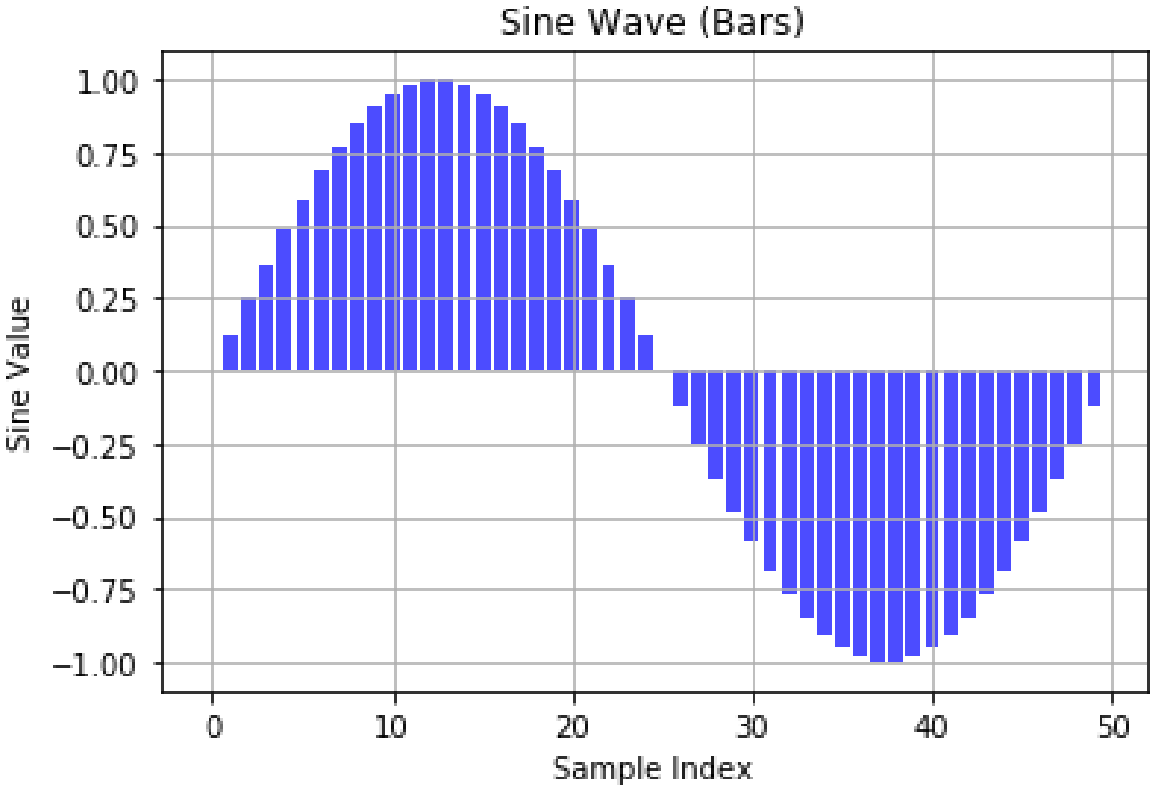
```
import math
nsamps = 50  # samples to generate
def generate():
    tau = math.pi * 2
    angle = tau / nsamps
    for i in range(nsamps):
        samp = math.sin(angle * float(i))
        print(f"{samp:.8f}")

generate()
```

```
 0.00000000
 0.12533323
 0.24868989
 0.36812455
 0.48175367
 0.58778525
 0.68454711
```

Plotting this, we get:



If we plot with bars,



We can actually turn this into something playable as a raw audio file.

Sample Rate : When we store sound digitally, we do it in tiny pieces called samples. The sample rate is like a measure of how many of these pieces we take every second. For example, a CD-quality recording takes 44,100 samples per second. When we refer to pieces or samples, we are essentially talking about numerical values that represent the amplitude (loudness) of the sound at specific points in time. Each sample is a discrete numerical value. These numerical values are then used to

recreate the original sound when played back. The sample rate determines how many of these numerical values (samples) are taken per second.

Frequency Range : Each sample represents a possible tiny moment in time. The sample rate also determines the highest pitch (frequency) of sound we can capture. In the case of CD-quality, it can go up to 22.05 thousand cycles per second (or 22.05 KHz). This range is enough because the human ear typically hears sounds between 20 cycles per second (Hz) and 20 KHz.

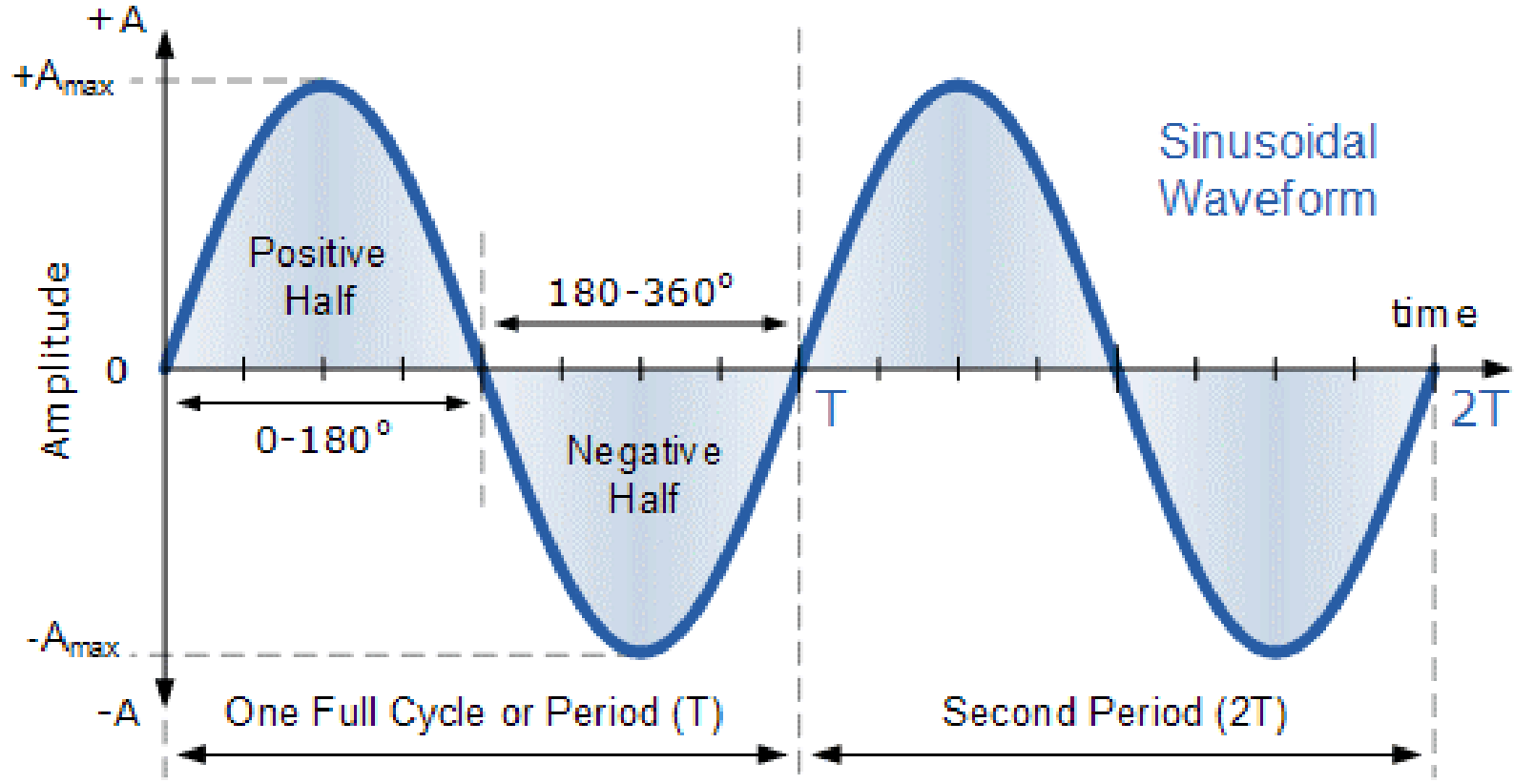
There are other formats with different sample rates, like 48 KHz for DVD-Video or 96 KHz for DVD-Audio. For simplicity, let's stick to CD-Quality (44.1 KHz) for now. Instead of using a small number of samples (like 50), which is fine for a simple example, for realistic audio, we'd need many more samples. For CD-quality, we'd need at least 44,100 samples to capture one second of sound accurately. To control how long the sound is, we introduce a variable that represents the duration. The duration is essentially how many seconds of sound we want to generate.

```python
import math
import struct

def generate():
    sample_rate = 44100  # CD-quality sample rate
    duration = 5  # Duration of the sound in seconds
    nsamps = int(duration * sample_rate)
    tau = math.pi * 2
    frequency = 440.0  # Example frequency, you can change it

    angle = tau / float(nsamps)
    file_path = "out.bin"

    with open(file_path, "wb") as f:
        for i in range(nsamps):
            sample = math.sin(angle * frequency * float(i))
            buf = struct.pack("<f", sample)
            bw = f.write(buf)
            print(f"\rWrote: {bw} bytes to {file_path}", end='')

# Call the generate function
generate()
```

We can play the audio once we got the bin file converted into mp3 or other formats.

All About WaveForms

A waveform is a graphical representation of a signal in the time domain, where the x-axis represents time, and the y-axis represents the amplitude or displacement of the signal.



TIME: The horizontal axis of the waveform represents time, typically measured in seconds. It shows how the signal evolves over time.

DISPLACEMENT: Displacement refers to the variation in air pressure caused by the movement of particles in a medium (such as air). As a sound wave travels through the air, it creates regions of compression (higher pressure) and rarefaction (lower pressure), leading to the perception of sound.

AMPLITUDE: Amplitude is the maximum displacement or distance from the equilibrium position of a wave.

FREQUENCY: Frequency is the number of oscillations or cycles of a waveform that occur in a unit of time. It is typically measured in Hertz (Hz), where one Hertz equals one cycle per second.

HOW SOUNDS WORK

Sounds are produced by the vibration of an object, such as a guitar string or a vocal cord. These vibrations create pressure variations in the surrounding medium (usually air) that propagate as waves. The characteristics of these waves, including amplitude and frequency, determine the perceived qualities of the sound.

PITCH & FREQUENCY

Higher frequencies result in higher-pitched sounds, while lower frequencies produce lower-pitched sounds. For example, a high-pitched whistle has a higher frequency than a low-pitched drumbeat.

LOUDNESS & AMPLITUDE

The amplitude of a sound wave determines its loudness. A larger amplitude corresponds to a louder sound, while a smaller amplitude results in a quieter sound.

HARMONICS

The shape of a waveform refers to the curve of the waveform line; in other words, how the displacement changes over time. We've been looking at a sine waveform. It is known as the fundamental waveform. This is because it is pure. There are no side effects. When we play a 440Hz sine wave, the only frequency we hear is 440Hz. Sine waves are the vanilla waves. When a waveform has side effect frequencies, we call them harmonics. Harmonics are always a multiple of the root frequency. Different waveforms have different selections,

but they always follow the same pattern.

- Fundamental note (root frequency): 1.6Hz.
- Second harmonic (2x frequency): 3.2Hz.
- Third harmonic (3x frequency): 4.8Hz.
- Fourth harmonic (4x frequency): 6.4Hz.
- This pattern continues to infinity.

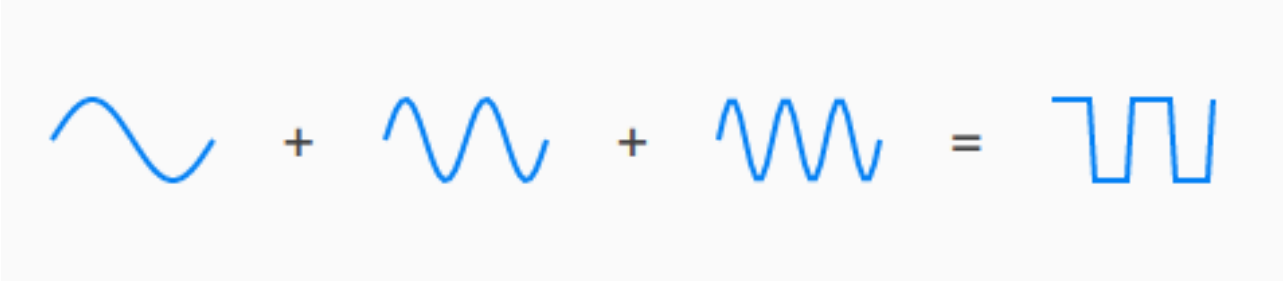
Triangle waves only have odd harmonics. That means they have the root note, 3rd harmonic, 5th harmonic, 7th harmonic, and so on.

Additive Synthesis

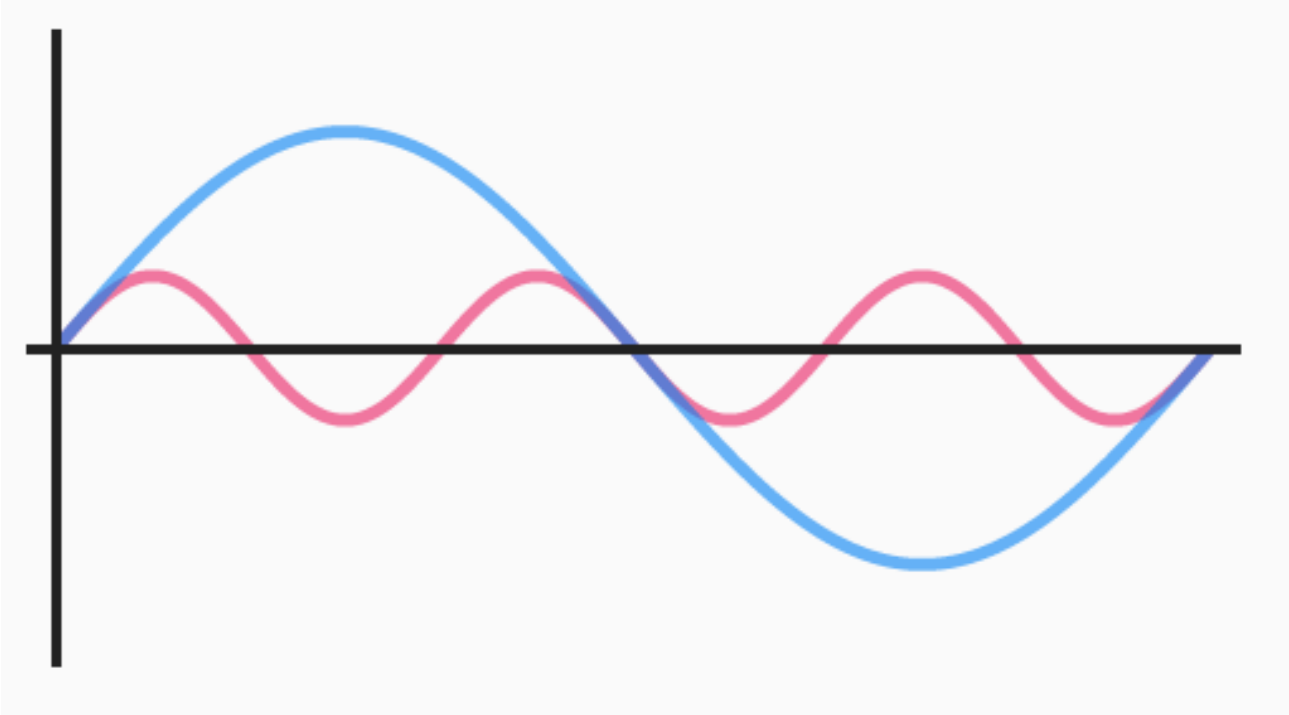
Imagine we have a simple sound, like a "beep" at a certain pitch. This sound is like a basic sine wave. It's like blowing a pure whistle - just one smooth note. Now, think about other sounds, like a musical instrument or a human voice. These sounds are more complex and not as smooth as a simple beep. When we look closely at them, it's like they are made up of several different whistles at different pitches, or frequencies, all combined.

Eventhough a sound might seem complicated, scientists found that we can recreate it by adding together a

bunch of those simple, smooth whistles (sine waves) at different frequencies. It's like playing multiple whistles at the same time to create a more complex sound. So, when we see a square wave or any other waveform, it's like looking at a musical recipe. It's made by combining many different smooth whistles (sine waves) at various frequencies. This is why different waveforms have different tones and characteristics.

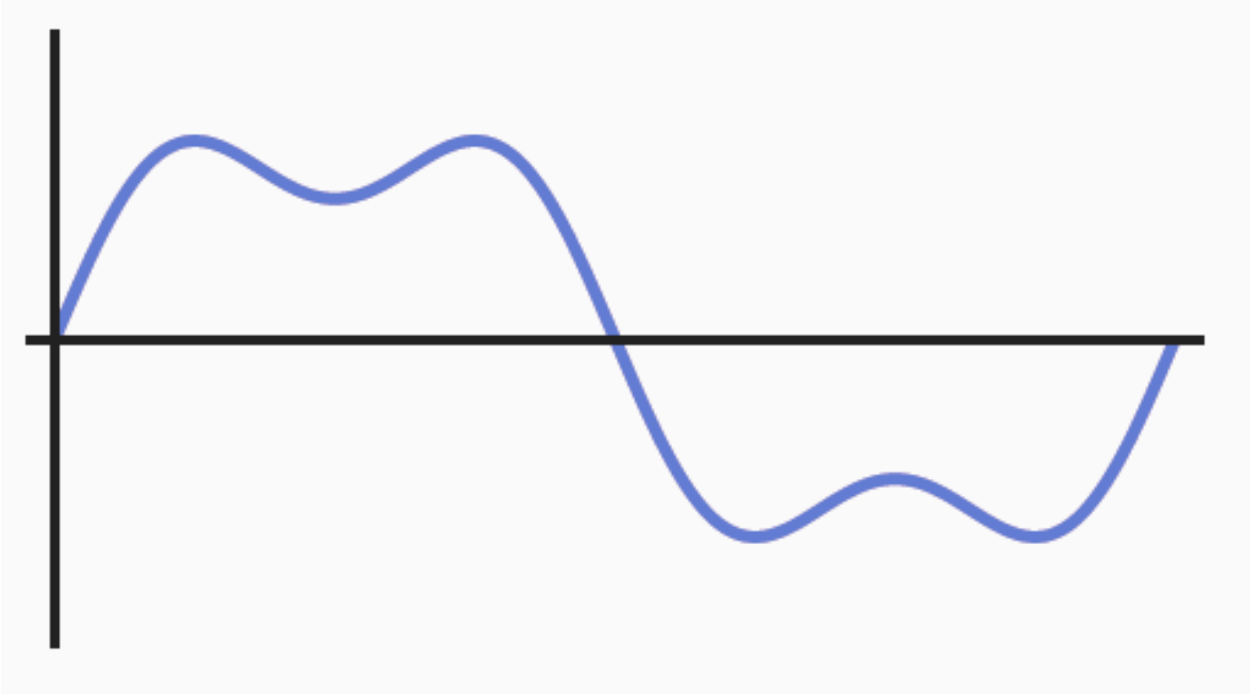


Say, we have two sound waves, one with 1Hz at 1 amplitude and another with 3Hz at 0.33 amplitude.



We just need to add the two waveforms together to get our end result. How does the addition work? It's arithmetic! The waveform graph is a bunch of individual points. At each point, we simply add the individual displacement values.

The new set of points is our new single waveform.

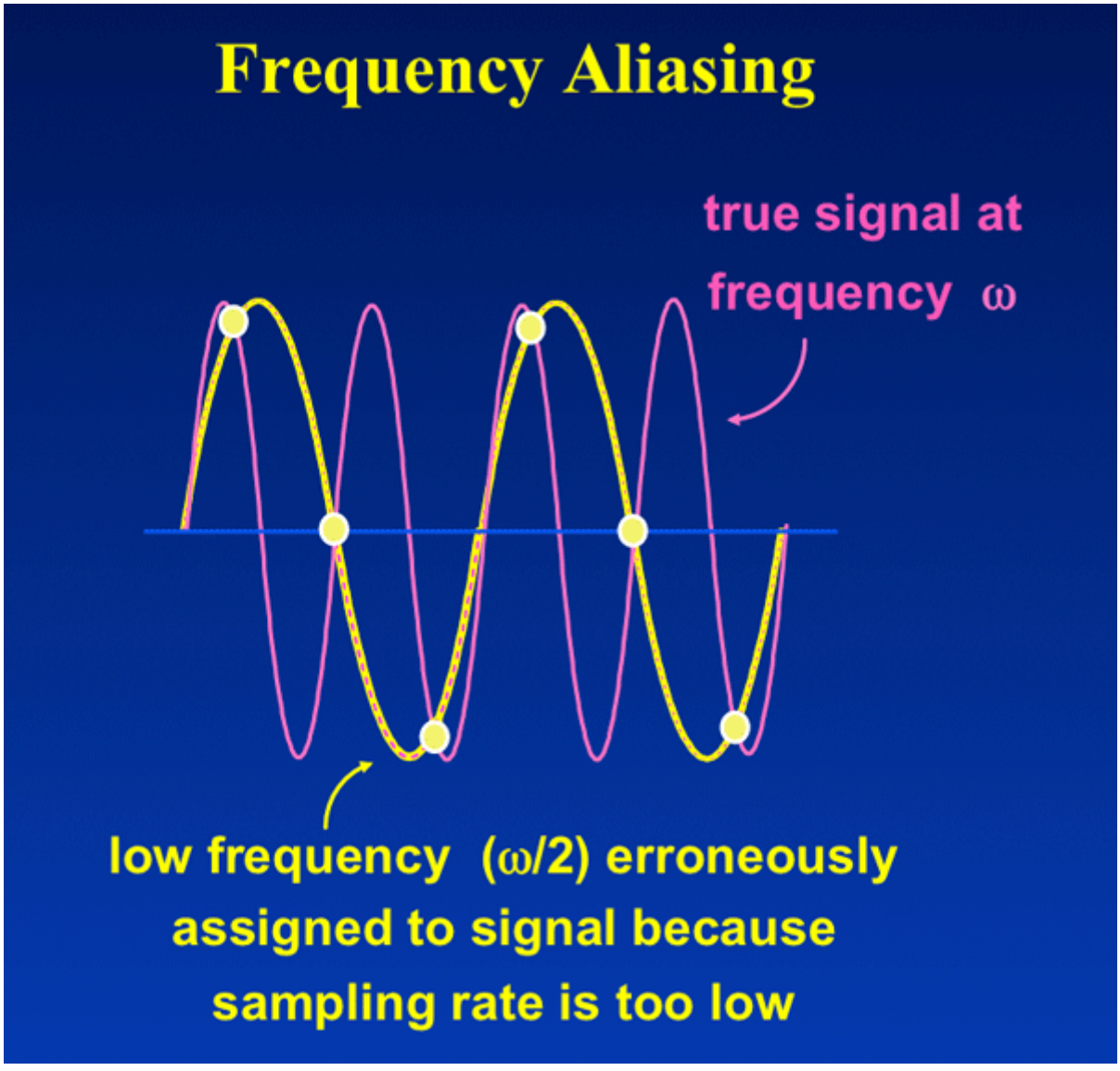


Nyquist Rate

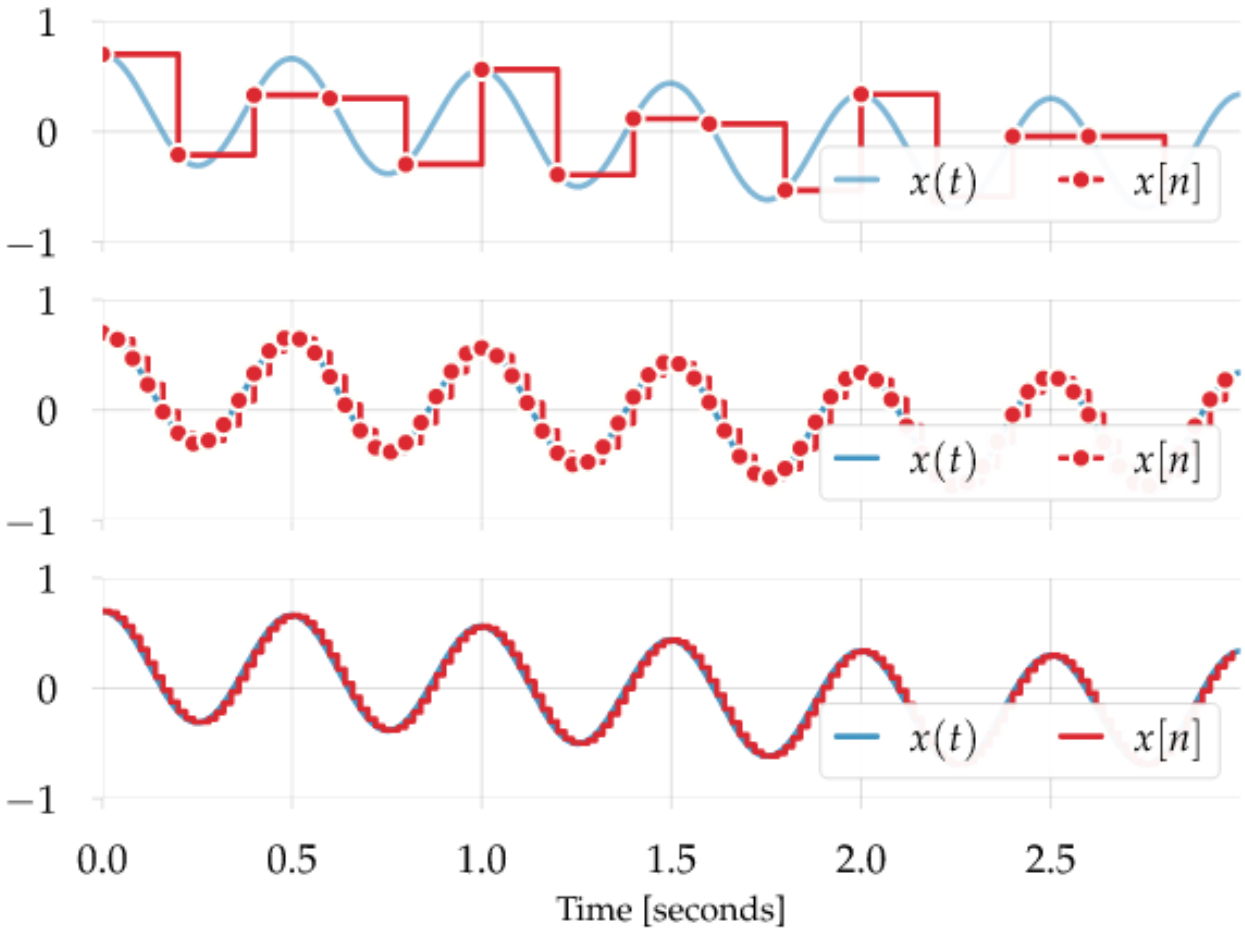
The Nyquist rate is the minimum rate at which a signal should be sampled to accurately reconstruct the original signal without introducing aliasing. Alias refers to the phenomenon where a signal is sampled, resulting

in distorted and misleading representations of the original signal. The Nyquist rate is defined as twice the highest frequency component present in the signal. Mathematically, it can be expressed as:

Nyquist Rate = 2.MAX(Frequency)

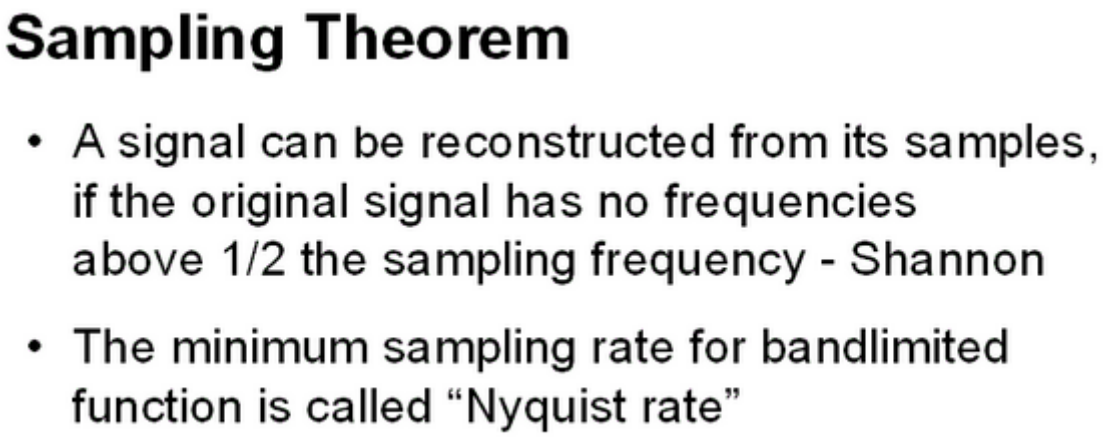


If a signal is sampled at a rate lower than the Nyquist rate, it can lead to aliasing, where higher-frequency components of the signal fold back into lower-frequency components, causing confusion and errors in signal reconstruction.



Sampling Theorem

The Nyquist sampling theorem states that to accurately reconstruct a signal from its samples, the sampling rate must be at least twice the highest frequency present in the signal. If we have a signal with frequencies up to a certain limit (let's call it the Nyquist frequency), we can sample it at a rate that is twice that frequency, and we won't lose any information.



HIGH FREQUENCIES

High-frequency components refer to variations in the signal that occur rapidly over time or space. Imagine a smooth gradient in a picture, such as a gradual transition from dark to light. The changes in intensity occur slowly across the image. On the other hand, if you have a pattern with rapid changes, like a fine texture or a detailed pattern, these variations represent high frequency components. Such high frequency components involve rapid changes or fluctuations, while low-frequency components involve slower variations.

BANDLIMITING

If your music has a lot of low bass notes and you set a cut-off frequency of 100 Hz, it means only sounds below 100 Hz will come through, and anything higher won't. It's like saying, "I only want to hear the really deep sounds." We use cut-off frequencies to focus on specific parts of a signal. It's like putting blinders on a horse, so it only sees what's right in front. In electronics or signal processing, we use cut-off frequencies to clean up signals or focus on certain aspects. If you band-limit a signal between 500 Hz and 2000 Hz, you're saying, "I only want to keep the sounds between these frequencies." Everything else gets

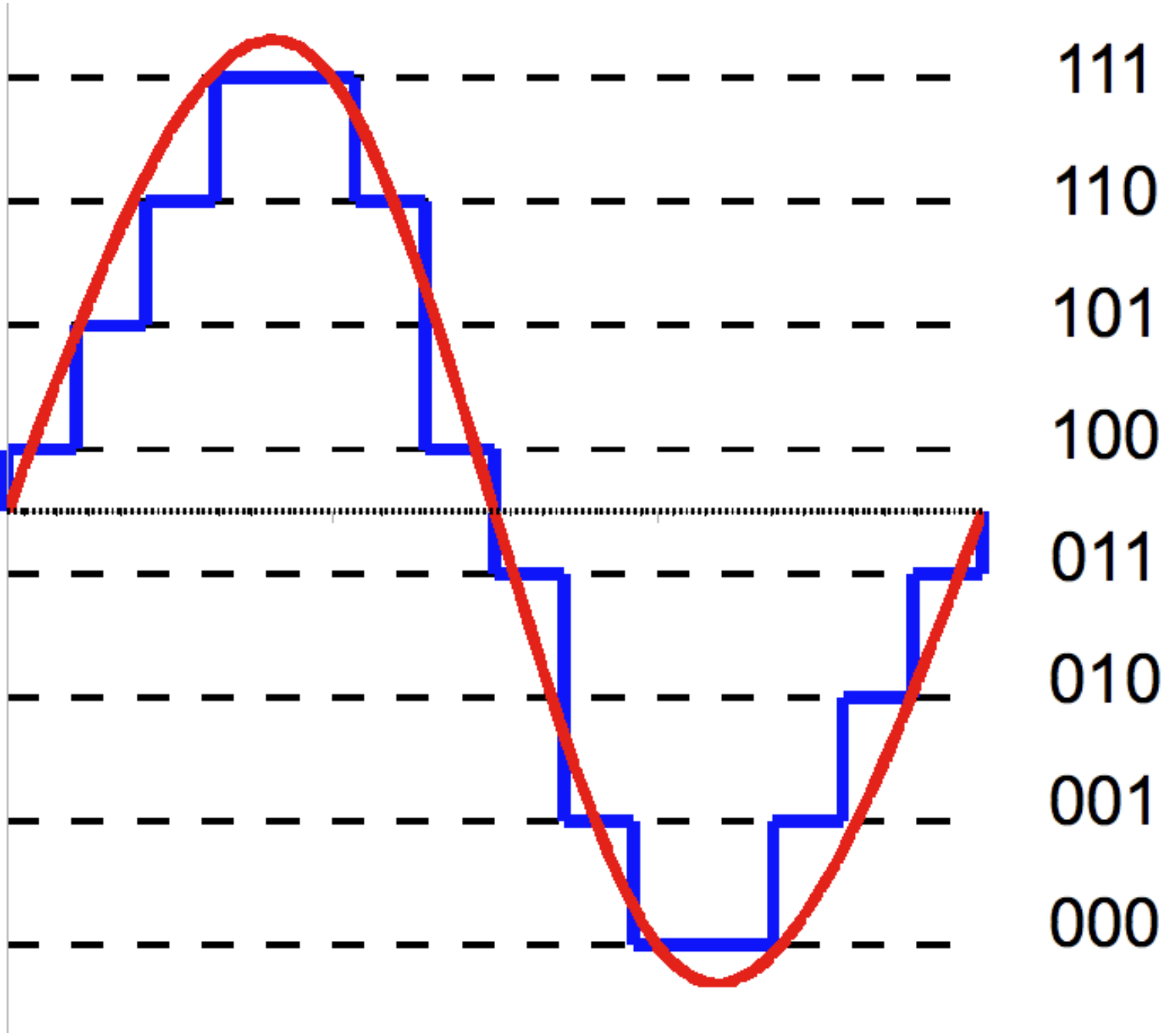
tossed out. Just like choosing specific channels on TV or radio, bandlimiting helps us focus on a particular range of frequencies. Bandlimiting in sampling is essential to prevent aliasing, a phenomenon that occurs when high-frequency components in a signal are incorrectly represented as lower-frequency components. When we sample a continuous signal to convert it into a discrete signal (digital signal), we need to ensure that the original analog signal is accurately reconstructed from the sampled values. The Nyquist-Shannon sampling theorem states that to avoid aliasing, the sampling frequency must be at least twice the

highest frequency present in the signal. If we fail to meet this criterion, high-frequency components can fold back into the lower frequency range, causing distortion and inaccuracies in the reconstructed signal. Bandlimiting involves filtering the signal before sampling to ensure that only the frequency components within the desired bandwidth are present in the signal. This is typically done using an anti-aliasing filter, which attenuates frequencies beyond the Nyquist limit. By limiting the bandwidth of the signal before sampling, we prevent high-frequency components from causing aliasing.

Quantization

Quantization refers to the process of representing continuous analog audio signals in a digital form by assigning discrete numerical values to the amplitude of the signal at specific points in time. This is a crucial step in the digitization of audio, as it allows for the storage and processing of audio information using digital systems. Resolution refers to the precision or detail with which the amplitude of an analog signal is represented in its digital form. Resolution is directly tied to the bit depth, which represents the number of bits used to represent the

amplitude of each sample in a digital audio signal. Common bit depths include 8-bit, 16-bit, 24-bit, and 32-bit.



The higher the bit depth, the greater the resolution and precision of the digital audio representation The number of quantization levels is 2 to the power of the bit depth. For example, with 16-bit audio, there are 2^{16} (or 65,536) possible quantization levels. More quantization levels mean finer distinctions in amplitude, resulting in higher resolution. Dynamic range refers to the difference between the quietest and loudest parts of a sound or piece of music. It represents the range of amplitudes or volume levels that a system can accurately reproduce. It is a crucial aspect of audio quality and

perception, as it affects the ability to capture and reproduce the full spectrum of sound, from the subtle nuances of quiet passages to the powerful impact of loud sections. Dynamic range is typically measured in decibels (dB), which is a logarithmic unit. The dynamic range is calculated as the difference in dB between the loudest and quietest parts of an audio signal. While higher resolution is desirable for audio quality, it comes at the cost of larger file sizes. More bits are required to represent each sample accurately, leading to increased storage and transmission

requirements. After quantization, audio files can undergo compression to reduce file sizes for storage or transmission. Lossy compression methods discard some audio data to achieve smaller file sizes, while lossless compression methods maintain all the original data. However, even lossless compression may not fully restore the original analog signal.

Quantization Error occurs in the process of converting continuous analog signals to discrete digital values during analog-to-digital conversion.

This error arises because the infinite set of possible analog signal values must be mapped to a finite set of digital values. The difference between the actual analog signal value and the nearest representable digital value is the quantization error. One of the most straightforward ways to reduce quantization error is to use a higher bit depth during analog-to-digital conversion. Higher bit depths provide more quantization levels, allowing for a finer representation of the analog signal and reducing the step size between digital values.

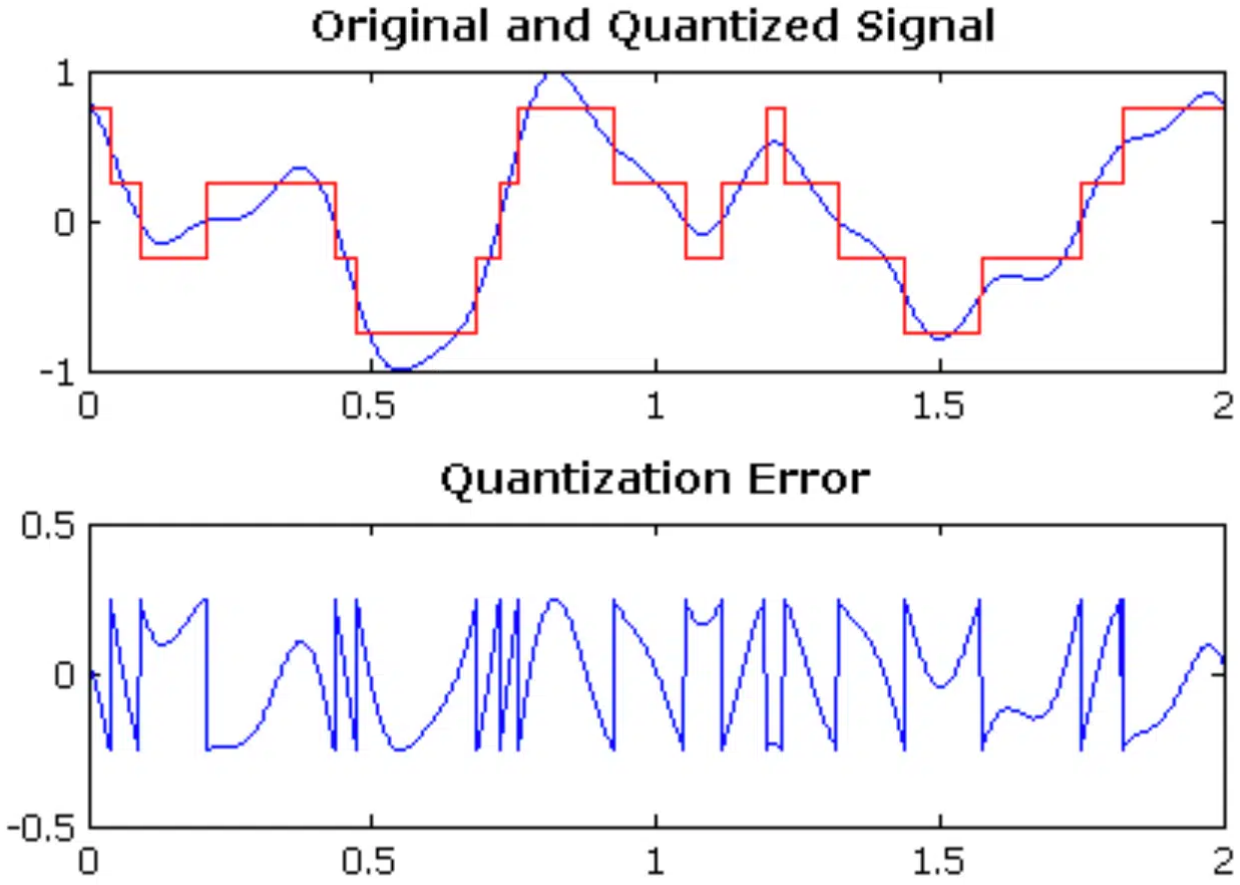
✺ Dithering is a technique that involves adding a small amount of random noise to the signal before quantization. This random noise helps spread out the quantization error, making it less perceptible. Dithering is particularly useful in low-amplitude portions of the signal, where quantization error might be more noticeable. In some cases, dithering shifts the quantization noise from being concentrated at specific frequencies to a more randomized distribution.

✺ Noise Shaping is a more advanced technique that involves shifting quantization noise to frequencies

where it is less audible. Noise shaping can improve the overall perceived audio quality by concentrating quantization noise in less audible frequency ranges. Some algorithms include, Pyschoacoustically Optimized Wordlength Reduction.

✺ Oversampling involves sampling the analog signal at a rate higher than the Nyquist rate before quantization. This higher sampling rate provides more information about the signal and can help reduce quantization error. However, oversampling may increase processing requirements.

✺ Sigma Delta modulation is a technique that uses feedback to reduce quantization error. It works by oversampling the input signal, converting the difference between the actual and predicted values to a 1-bit digital signal, and then using feedback to adjust the prediction.



Channels, Codecs

Audio is made up of samples that represent the amplitude of the sound wave at different points in time. These samples are stored in various formats within audio files, such as 16-bit integers, 32-bit floating-point values, or other formats. Each sample's size is referred to as the sample size. The position of a sound source in the audio signal is called a channel. Channels indicate where the audio is coming from, and in stereo sound, there are typically two channels: left and right speakers. The collection of channels in an audio signal is known as the channel count.

In multi-channel audio files, channels are organized into frames, each containing one sample for each channel. Stereo audio, with a left and right channel, is common on the web, often using 16-bit samples. For 16-bit stereo audio at a standard sample rate of 48 kHz, each second of audio takes up 192 kB of memory. Compression becomes essential due to the large memory requirements, and this is achieved through encoding and decoding with audio codecs. Audio codecs, short for COder/DEcoder, handle the compression and decompression of audio. Various codecs exist, with several commonly used on the web.

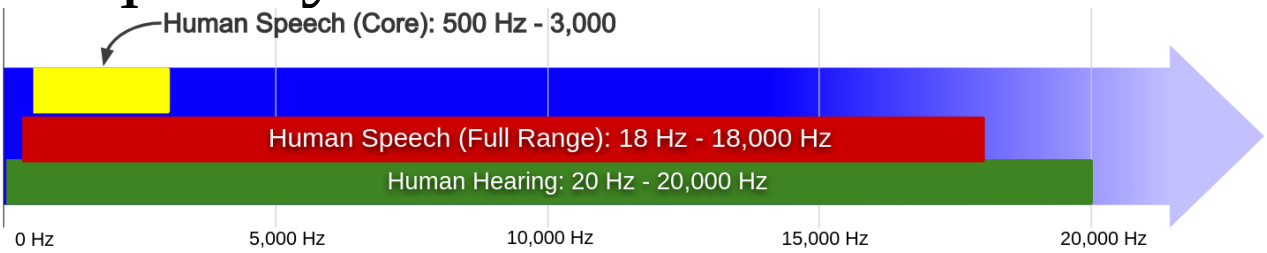
Audio Compression

Traditional compression tools like zip struggle with audio because they rely on replacing repeating patterns with shorter representations. To compress audio, various techniques are used. Many codecs combine these methods and may incorporate additional ones. One simple approach is applying a filter to eliminate hiss and quiet sounds, turning quiet parts into silence. This can create silent stretches and repeating signals that can be shortened. Another technique involves narrowing the audio bandwidth, removing frequencies deemed unnecessary. This is particularly effective for voice-only

audio, reducing data and making the signal more compressible. Audio data, unlike text or other types of data, is often not precisely repeated, making it challenging to compress using standard algorithms like those in zip files. Compression techniques for audio involve understanding psychoacoustics, the science of how humans perceive sound. Compression methods consider factors like our sensitivity to frequency changes, the range of human hearing, and the importance of certain frequencies in different contexts. By leveraging psychoacoustics, compression algorithms aim to minimize the compressed audio size while

preserving perceived sound fidelity. When choosing a compression codec, the fundamental question is whether it's acceptable to lose some audio fidelity upon decoding. Lossless codecs preserve all details but result in larger file sizes, while lossy codecs sacrifice some details for significantly smaller files. Lossy codecs, the more common choice, use psychoacoustic analysis to discard less noticeable components of the audio waveform. They also apply various algorithms and transforms to simplify and reduce audio size. Lossless codecs, on the other hand, maintain precise audio reproduction but offer less room for optimization.

Parameters for configuring lossless encoders involve selecting encoding algorithms and specifying the level of processing allowed. Lossy encoders have options to balance quality and size, including bit rate control and adjustments to audio frequency bandwidth.



Understanding the characteristics of the audio content helps determine the choice of codec. Speech, with a narrower frequency range, is more easily compressed than complex music waveforms. Codecs may discard frequencies outside the human hearing range, further reducing data size without

significantly impacting perceived quality. Joint stereo coding, like mid-side and intensity stereo coding, optimizes storage by exploiting similarities between left and right channels. While these methods introduce some loss, the impact is often imperceptible to the average listener. The choice between lossless and lossy compression depends on the desired audio fidelity and the specific use case, considering factors such as available storage, network bandwidth, and audience expectations. Codec parameters and techniques like joint stereo coding play key roles in optimizing the balance between file size and sound quality.

Audio Encoding

Audio encoding involves the storage and transmission of audio data in digital form. It's important for understanding how the Speech-to-Text API processes audio.

Format vs. Encoding : A file format like .WAV defines the header of an audio file but not its encoding. For example, a .WAV file may use linear PCM encoding. FLAC serves as both a file format and an encoding.

FLAC Header : FLAC requires a header in audio data. Other encodings, like LINEAR16, do not need headers.

Automatic Detection: Speech-to-Text can automatically detect encoding

and sample rate for WAV or FLAC files based on the file header.

Supported Audio Encodings

Codec	Name	Lossless	Usage Notes
MP3	MPEG Audio Layer III	No	MP3 encoding is a Beta feature and only available in v1p1beta1. See the RecognitionConfig reference documentation for details.
FLAC	Free Lossless Audio Codec	Yes	16-bit or 24-bit required for streams
LINEAR16	Linear PCM	Yes	16-bit linear pulse-code modulation (PCM) encoding. The header must contain the sample rate.
MULAW	μ-law	No	8-bit PCM encoding
AMR	Adaptive Multi-Rate Narrowband	No	Sample rate must be 8000 Hz
AMR_WB	Adaptive Multi-Rate Wideband	No	Sample rate must be 16000 Hz
OGG_OPUS	Opus encoded audio frames in an Ogg container	No	Sample rate must be one of 8000 Hz, 12000 Hz, 16000 Hz, 24000 Hz, or 48000 Hz
SPEEX_WITH_HEADER_BYTE	Speex wideband	No	Sample rate must be 16000 Hz
WEBM_OPUS	WebM Opus	No	Sample rate must be one of 8000 Hz, 12000 Hz, 16000 Hz, 24000 Hz, or 48000 Hz

Why Encode?

Digital encoding reconstructs analog waveforms using sampled amplitudes and precise bit depths.

A sound device's fidelity depends on its frequency response and dynamic range. Encoding efficiently stores and transports audio data. Compression techniques may compensate for smaller bit depths, but they tend to be lossy.

Best Practices:

Choose Lossless Formats: Use lossless encodings like FLAC or LINEAR16 for better speech recognition.

Avoid Lossy Formats: Lossy formats may degrade accuracy, so avoid them if you have control over the source audio.

Audio Formats

Many different audio file formats exist for storing recorded audio data on a computer system. There are three main types of audio file formats

1. Uncompressed

Uncompressed audio formats store the audio information as it is recorded. This results in big files, but no information is lost, therefore they are suitable for archiving original recordings. The most common uncompressed audio format is PCM, which is usually stored in a WAV or AIFF file.

2. Lossless compression

Lossless audio compression formats need less space then uncompressed formats, without any loss in quality. They work similar to ZIP, but the compression algorithms are specifically designed for audio data. Some example formats are FLAC (Free Lossless Audio Codec) or ALAC (Apple Lossless Audio Codec), for a comparison of various codecs see Lossless comparison.

3. Lossy compression

Lossy compression formats significantly reduce the file size, by throwing away information imperceptible to humans. This gives very small files, but some information is lost and cannot be reconstructed. The best-known example is MP3, others are AAC, Opus, Ogg Vorbis, WMA.

Bitrate

The Bitrate defines, how many bits (storage) are used to encode a certain amount of audio - for example, 128 kbps will use about 128 kilobits for each second of audio that is encoded. In Constant Bitrate Encoding (CBR), the bitrate is kept constant across the entire file, whereas Variable Bitrate Encoding (VBR) tries to maintain a constant quality by choosing the optimal bitrate to represent each audio frame (for example: a higher bitrate for complex audio, much lower bitrate for silence).

The quality of audio compression formats like MP3, WAV, and Opus can vary based on the bitrate used for encoding.

MP3

At lower bitrates, MP3 tends to exhibit noticeable loss of audio quality. MP3 quality improves with higher bitrates, providing a good compromise between file size and audio fidelity. At the highest MP3 bitrates, the loss of audio quality is minimal, and the difference from the original source may be difficult for most listeners to perceive.

WAV

WAV is a lossless format, meaning it retains the full audio quality of the original recording. It is not a compressed format, so file sizes are significantly larger compared to compressed formats like MP3 and Opus. WAV provides the highest audio quality but comes at the cost of larger file sizes, making it less practical for storage and streaming.

Opus

Opus is known for its efficiency at low bitrates. Even at very low bitrates, it can provide decent audio quality, making it suitable for applications with bandwidth constraints, such as online communication. Opus excels in maintaining good audio quality while keeping file sizes relatively small. It often outperforms other codecs like MP3 at comparable bitrates. At higher bitrates, Opus continues to provide excellent audio quality with relatively small file sizes. It may not match the lossless quality of WAV.

NANDRI