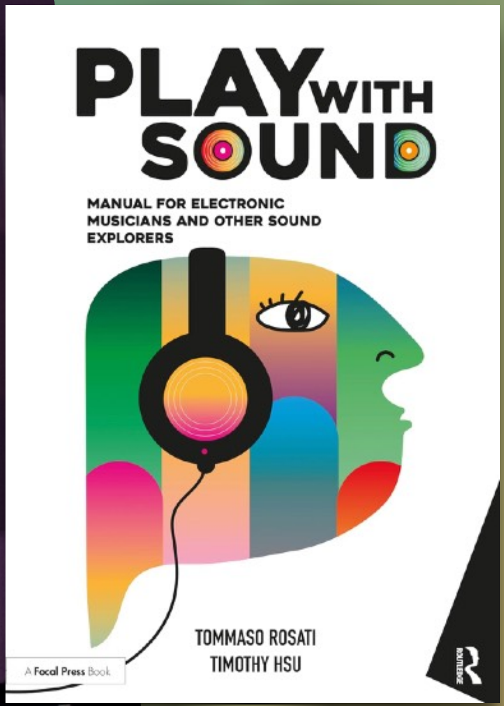
TOMMASO ROSATI
SOUND ART

SAMPLING AND QUANTIZATION

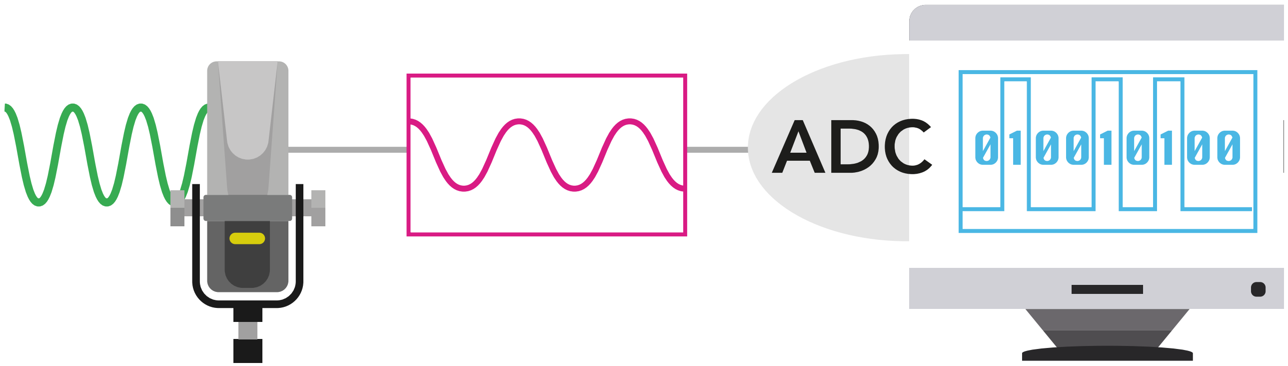
DIGITAL SOUND
SAMPLING
QUANTIZATION

THE BOOK IS NOW AVAILABLE!




ADC
010010100
DAC

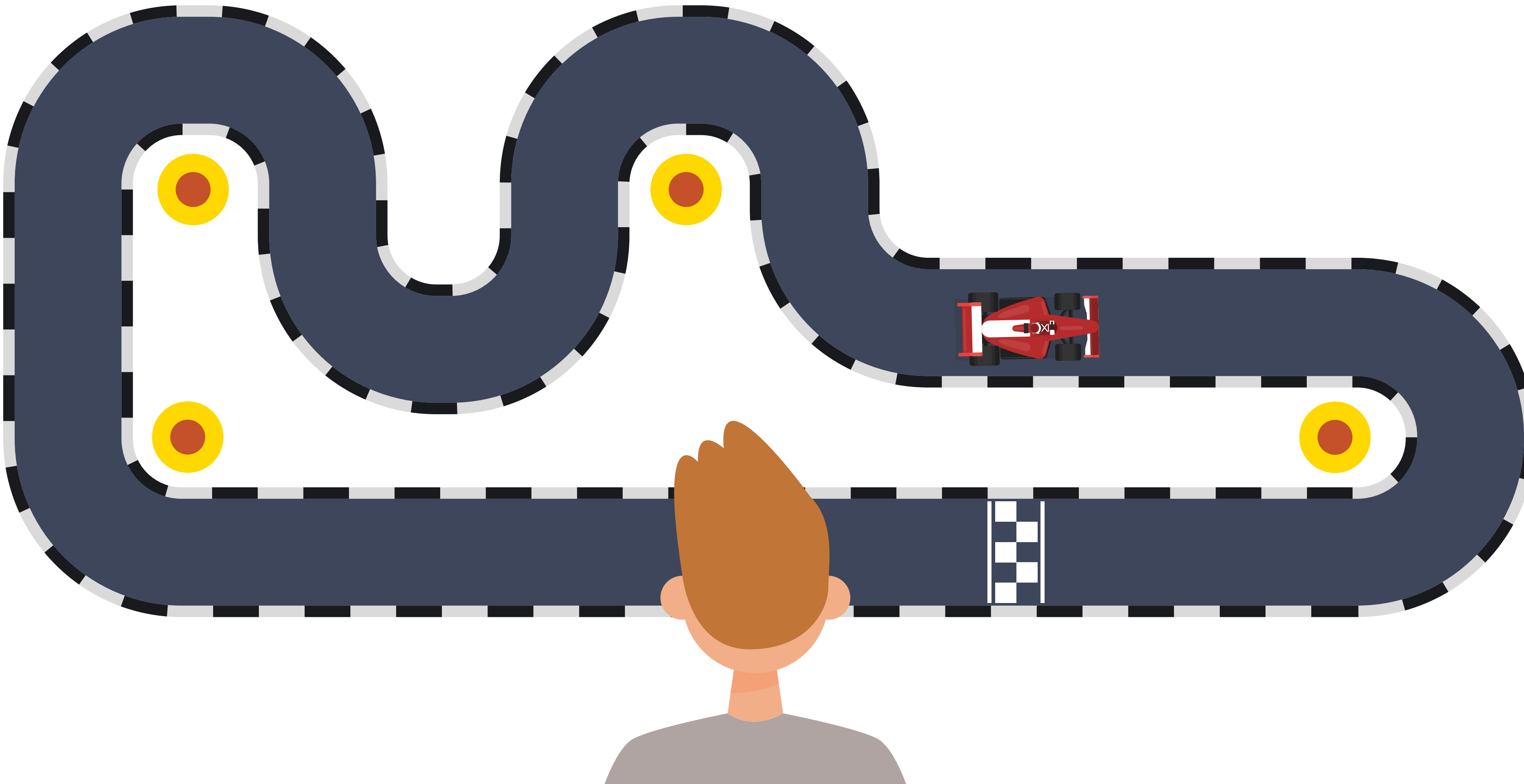
ADC



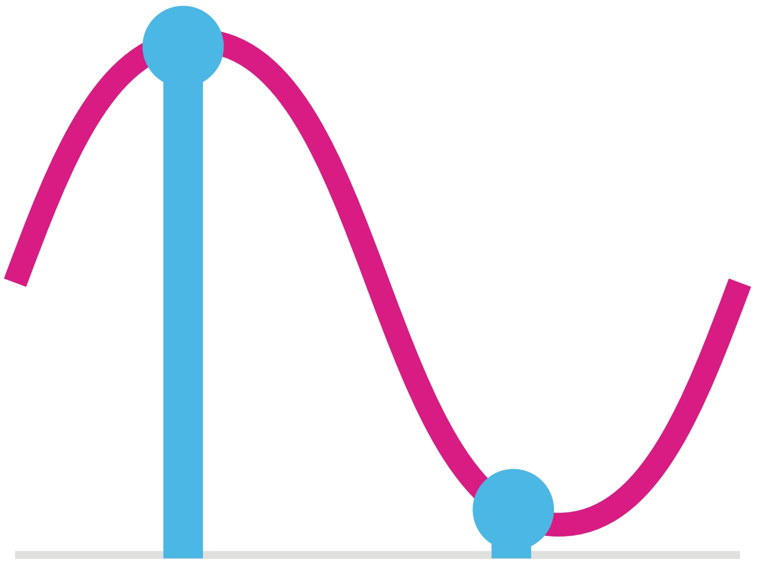
An analog wave has infinite points since it is a continuous signal. We must discretize and sample this signal in order to convert it to a digital signal.

But at what **frequency** should I take my samples in order to then reconstruct my original signal without loss of information?

ADC



ADC



1

For each wave cycle, I will need at least two sample points, evenly spaced, to represent the wave in the digital domain.

ADC

2

I consider the case where I want to be able to reproduce the highest frequency that we can hear, **20,000 Hz.** What should my sampling rate be?

To do this, since I need to take 2 points per cycle. Thus, I will have to sample at twice that frequency: **40,000 Hz**.

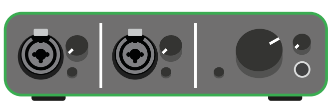
twice the bandwidth of frequencies that I want to represent

fs > 2B

sampling frequency

Nyquist-Shannon sampling theorem

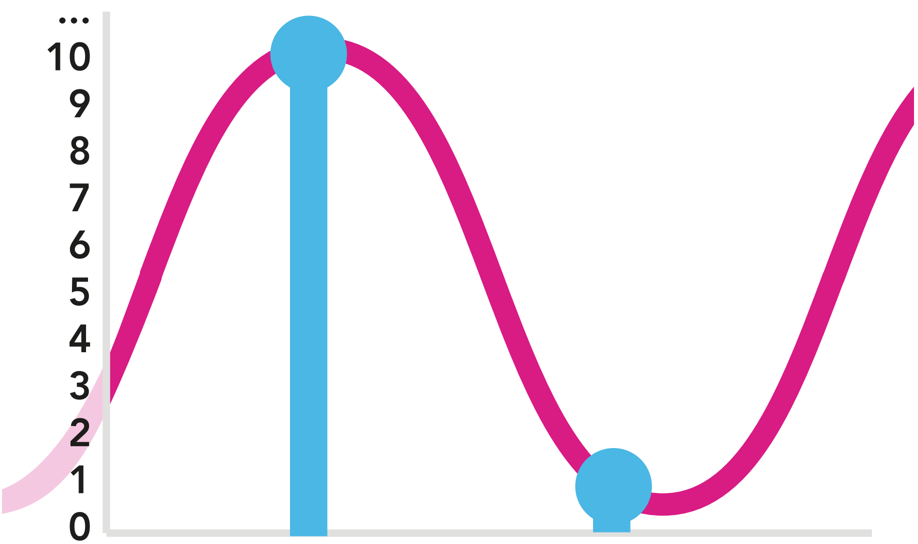
ADC

	CD	44100 Hz	16 bit
	VIDEO	48000 Hz	16/24 bit
	AUDIO INTERFACE	96000 Hz	24 bit

ADC

I have determined the sampling frequency, but what amplitude values should I assign my samples?

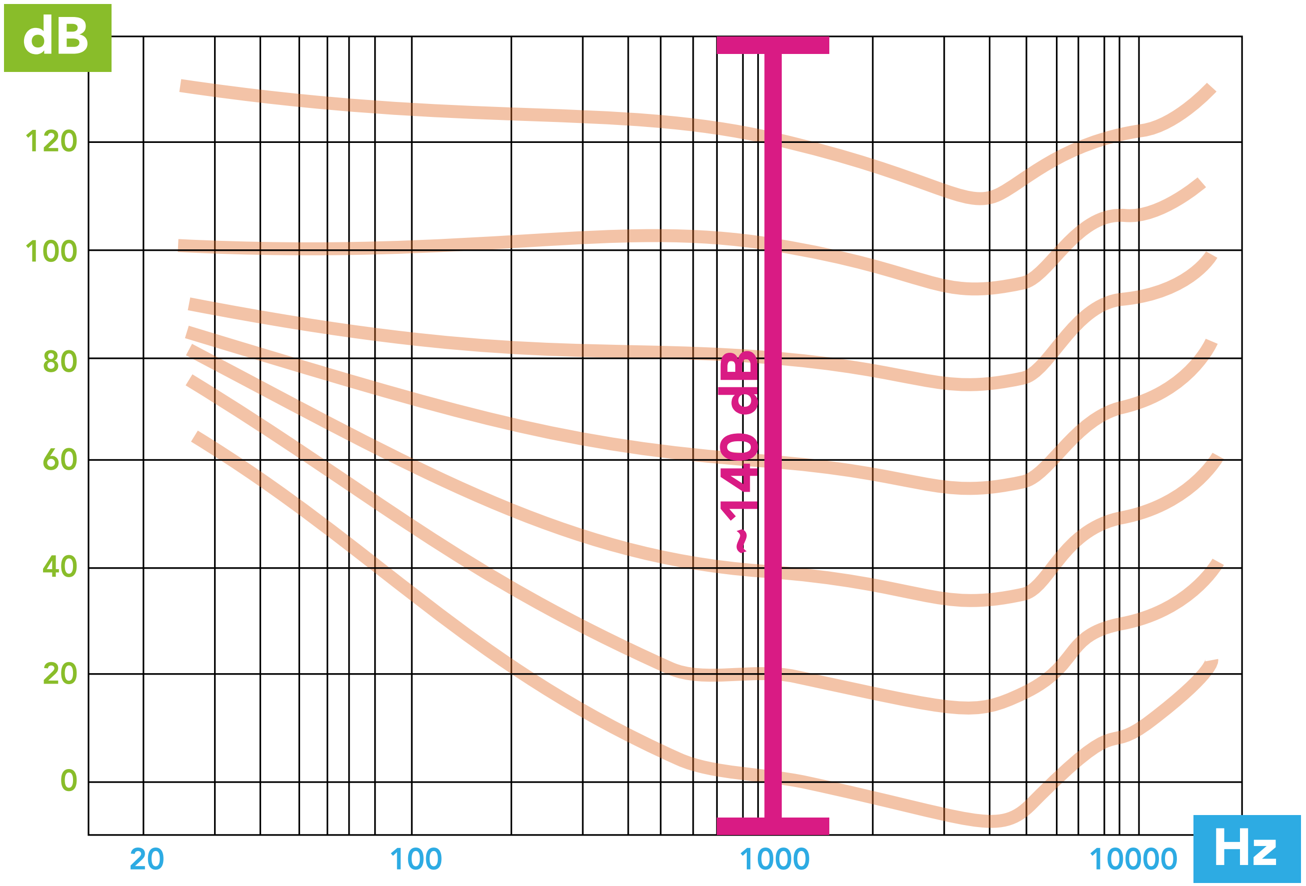
How many amplitude steps do we need to sufficiently characterize audio waveform? Quantization is the process by which
we convert these continuous amplitude values into discrete amplitude steps.



ADC

1

The maximum range that our auditory system can perceive is 140 dB relative to a reference of 0 dB, which is perceptual silence.



ADC

2

In digital, data is stored in bits, which can assume a state of **0** or **1**.

We need to determine how many bits we need to correctly represent 140 dB of dynamic range, considering that each individual bit has two possible states, 0 or 1.

To calculate how many total states are possible with 2, 3, 5, 16, or N number of bits, we compute 2^N. In laypersons terms, we raise 2 (the number of possible states for each bin) to the number of bits we have in our system.

$$2^N$$

possible states of 1 bit (0 or 1) — **2**

N — **number of bits**

ADC

3

In dB, doubling the amplitude results in an increase of 6 dB because decibels are on a logarithmic scale.

If we only have 1 bit, there are only two possible states: 0 or 1. I consider 0 as silence and 1 as sound. I can represent 6 dB of sound with 1 bit.

And if I now want to double the number of possible states, I need to increase my bit depth by one bit.

With 2 bits, I have 4 possible states, so I can represent 12 dB (double the amplitude of 6 dB)

With 3 bits, I have 8 states and can represent 18 dB, a 6 dB increase (or doubling) from before.

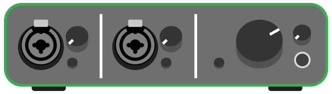
Etc.

ADC

+1 bit = a gain of 6dB in signal-to-noise ratio

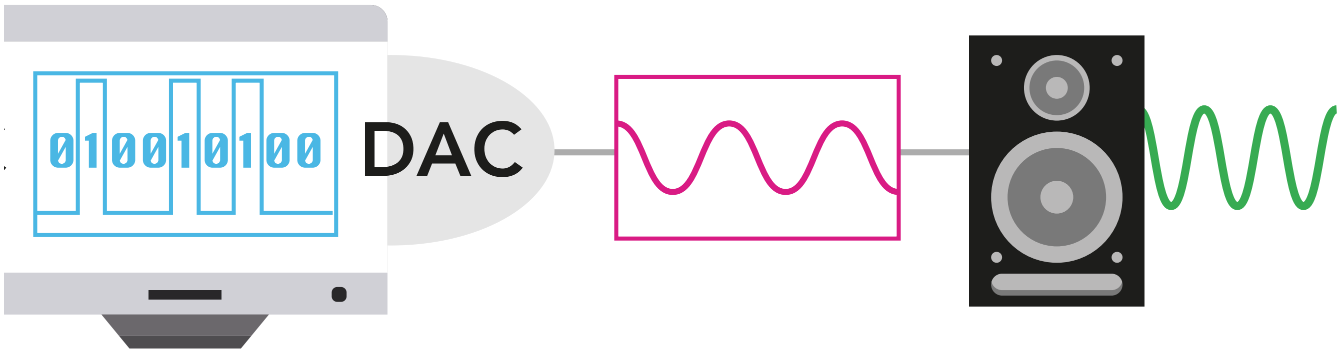
bit	states	signal-to-noise ratio (dB)
1	2	6
2	4	12
3	8	18
4	16	24
5	32	30
8	256	48
16	65,536	96
24	16,777,216	144

DAC



CD	44100 Hz	16 bit
VIDEO	48000 Hz	16/24 bit
AUDIO INTERFACE	96000 Hz	24 bit

DAC

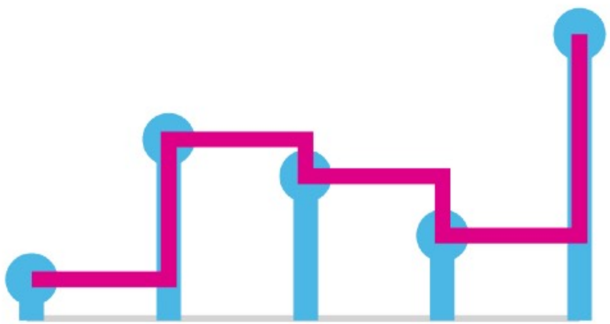


In the phase of reconstructing the analog signal, I have a digitally stored number that I need to turn back into a continuous wave. I will need to fill in the data gaps between the samples during reconstruction.

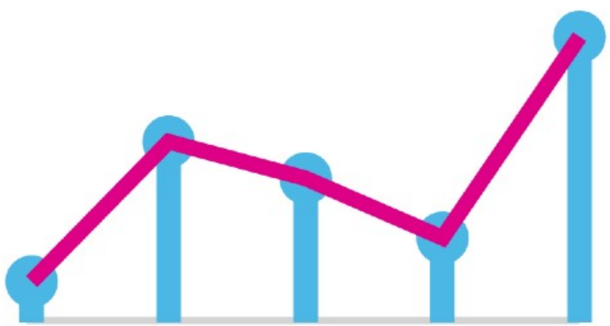
What methods can interpolate between the samples in order to faithfully bring back the original continuous sound?

DAC

Historically, various methods have been used before arriving at the most accurate function currently used:

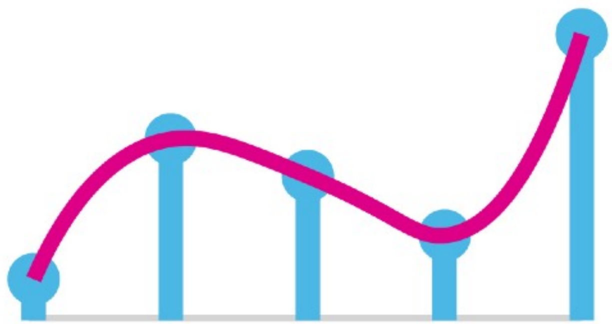


Sample and Hold
takes the value of a sample and holds that value until we get to the next sample. However, these abrupt and sharp "steps" can create unwanted harsh timbres.



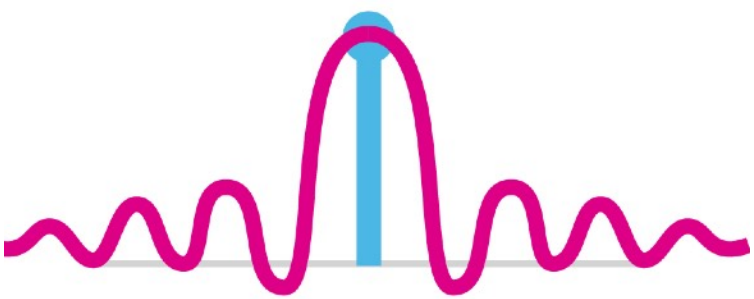
Linear interpolation
directly connects one sample's value to the next sample's value using a straight line with a constant slope. This method also creates sharp angles that can result in a harsh sound.

DAC



Cubic splines

improves the result but is still not "perfect."

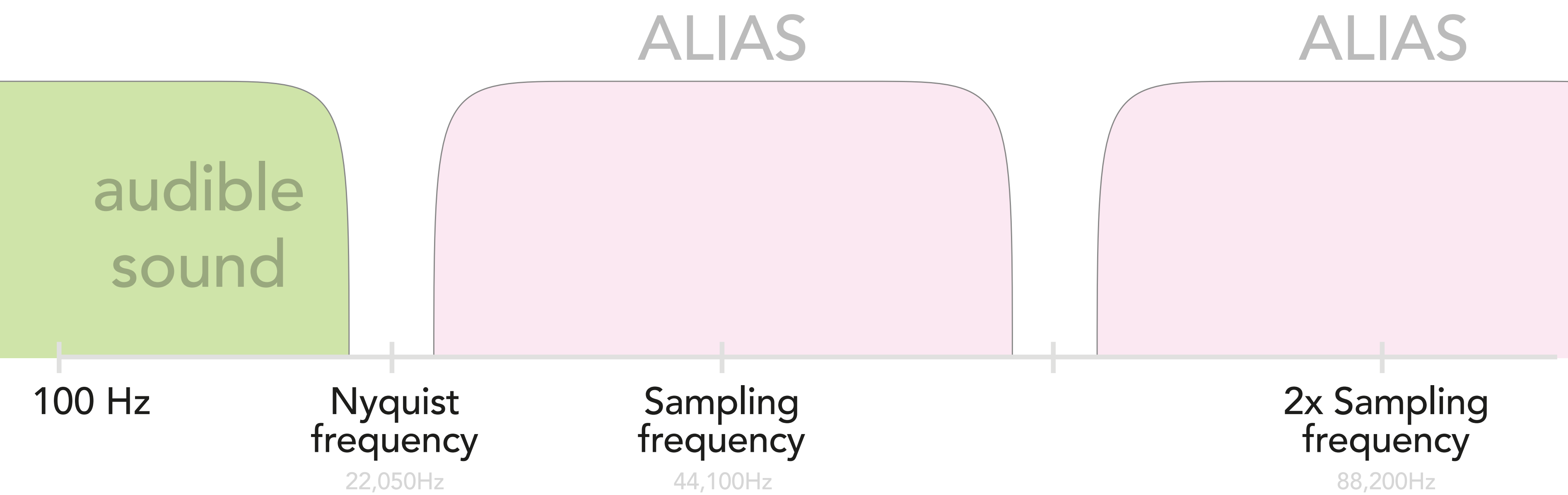


Cardinal Sine

This function reconstructs an analog signal that is virtually identical to the original wave.

ADC-DAC

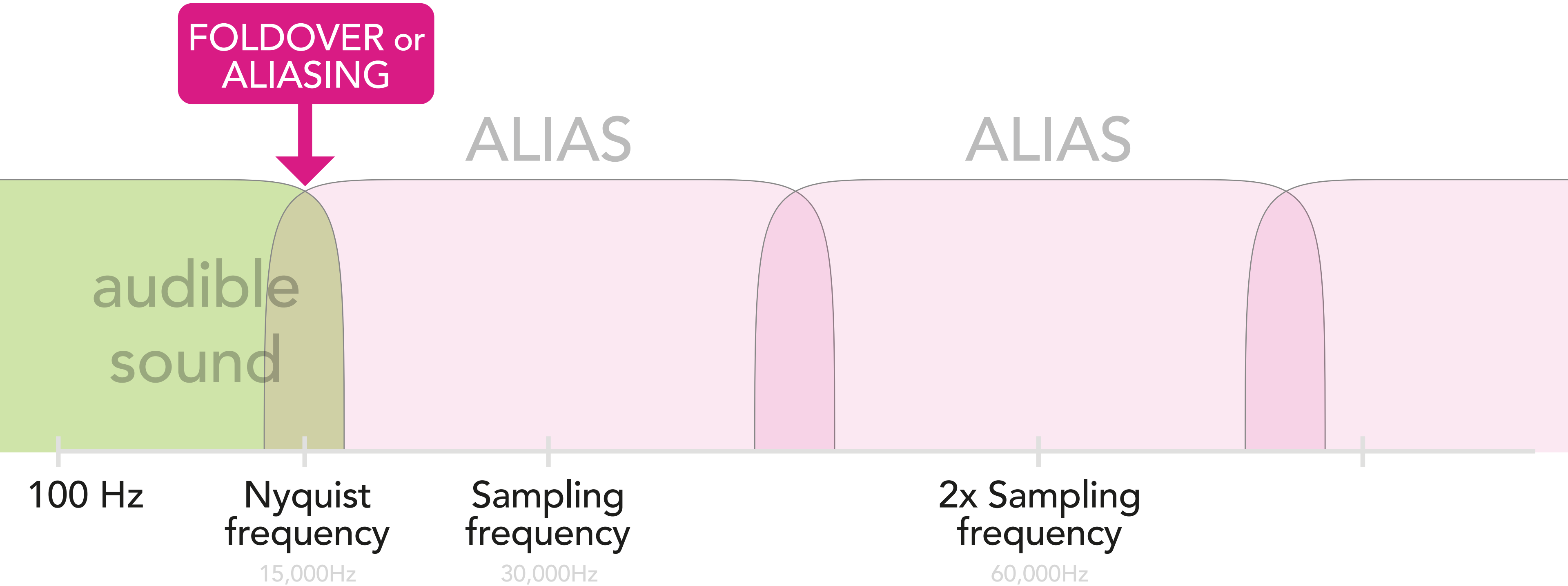
When I sample at a certain sampling frequency, ambiguities are created at multiples of the sampling frequency.
The ambiguity can be seen in the spectral copies of the original signal that occur. These copies are called **aliases**.



Generally, with an appropriate sampling rate, the ***aliases*** do not necessarily present too many problems as long as it occurs at very high frequencies that are out of the audible spectrum.

ADC-DAC

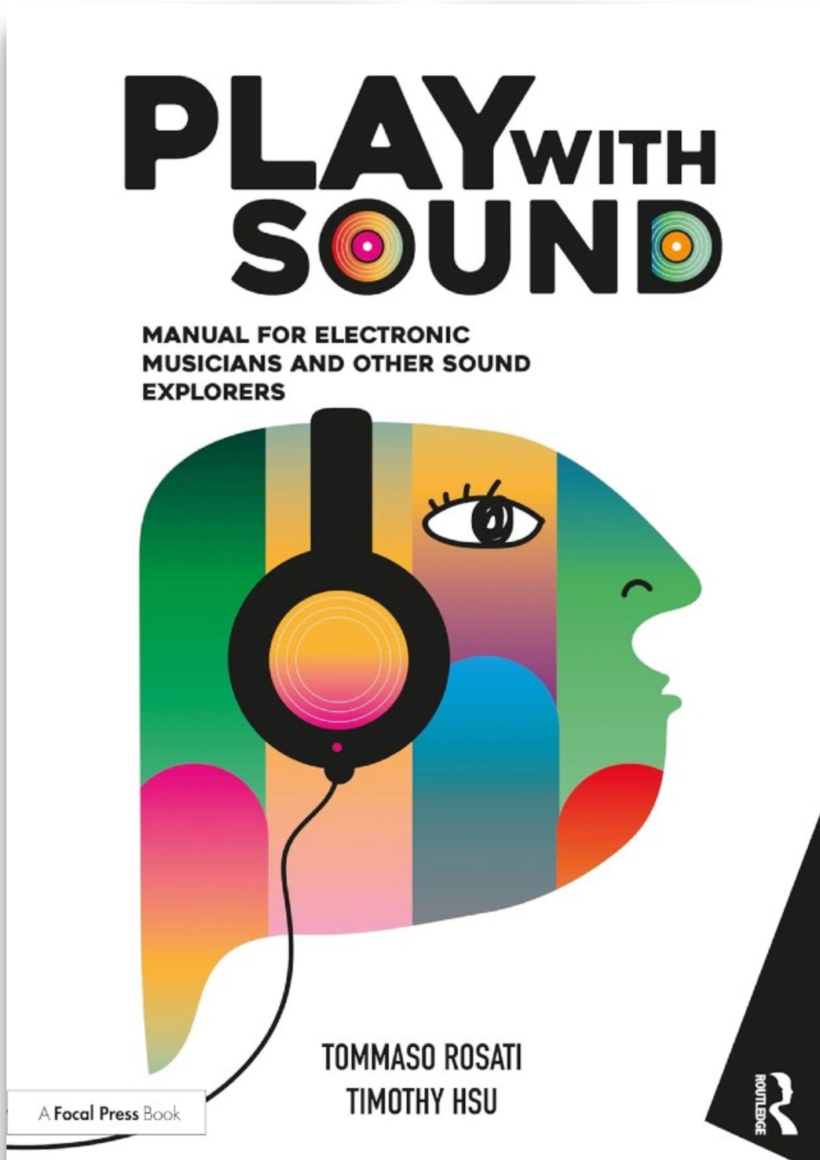
The phenomenon of **aliasing** starts to happen when the ***aliases*** spill over into the audible spectrum and overlap one another. The reconstructed sound is then "enriched" with frequencies that are not supposed to be there. This effect is known as **foldover** distortion, or **aliasing**, and occurs when the sampling rate is too low, or if we try to sample sounds that exceed the Nyquist frequency.



ADC-DAC

There are some methods during sampling or reconstruction that can help us avoid aliasing distortion:

- using **anti-aliasing filters**
  - at the input to prevent the contribution of frequencies that are too high
  - at the output, to attenuate any aliases occurring at very high frequencies
- increasing the **sampling frequency**
- Using **oversampling:** during reconstruction, adding artificial samples between real ones virtually doubles (or more) the sampling rate, helping to avoid foldover distortion.



THE BOOK IS NOW AVAILABLE!