4.5.6.1 Representing Sounds

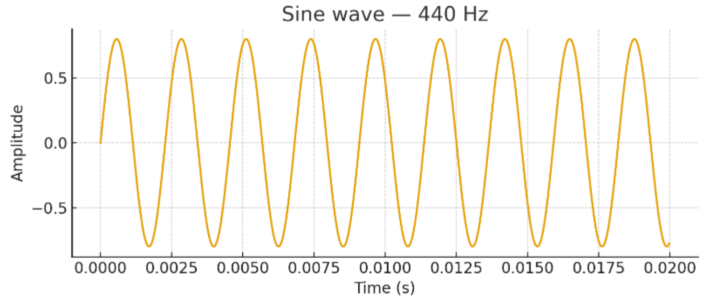
Bit Patterns

Bit patterns can be used to represent many other forms of data other than numerical values and characters. Computers also need to store signals and data to represent sounds and graphics.

Signals and data can be analogue or digital.

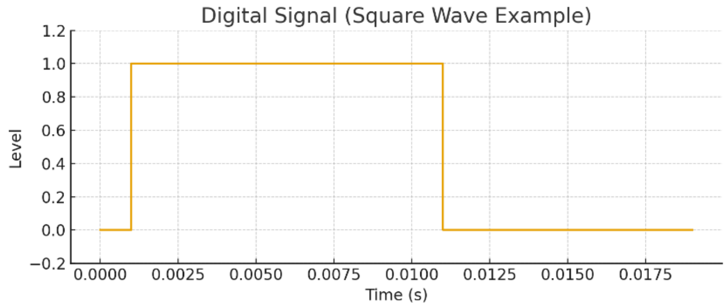
Analogue is continuous data that **can take any value** within a range.

E.g. Sound waves, temperature, light intensity



Digital is discrete data that **can only take specific values** (values that can be encoded in binary).

E.g. binary data in computers, digital images like bitmaps, digital sounds like MP3s.



ADC and DAC

Computers can only use binary data so analogue data must be converted to digital before it can be processed and stored.

An **analogue to digital converter (ADC)** takes an analogue signal as an input and converts it into a digital signal. It is used to convert sound waves into a digital format.

A microphone is an analogue sensor that can take an analogue signal for an ADC to convert into a digital signal.

A digital to analogue converter (DAC) takes a digital signal as an input and converts it into a analogue signal.

A computer would use a DAC to convert a stored MP3 to an analogue signal that could be output by a speaker.

Digital Representation of Sound

Converting analogue sound waves into a digital representation requires **taking samples of the amplitude of the wave at regular time intervals**.

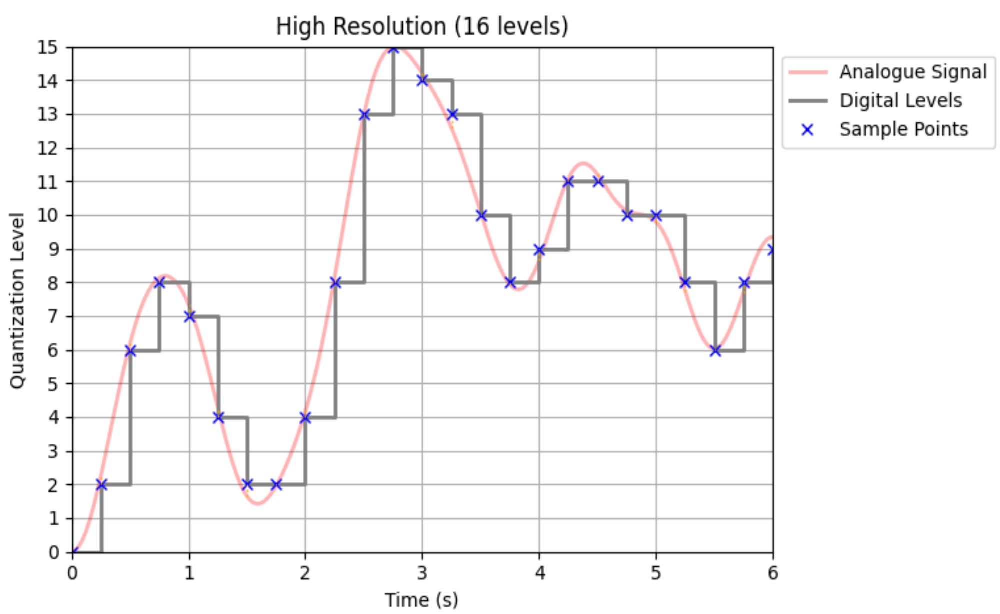
The amplitudes need to be encoded into binary values. The number of bits available for this encoding is called the **sample resolution**.

The number of samples taken per second is called the **sample rate** which is measured in hertz (Hz).

Example: An analogue signal (red) is being sampled to get a digital signal (grey).

This sample has 16 different levels that it can use to encode the amplitude level for a sample of the analogue wave. 16 different levels can be encoded using 4 bits. **The sample resolution is 4 bits.**

4 samples are being taken every second. **The sample rate is 4Hz**.

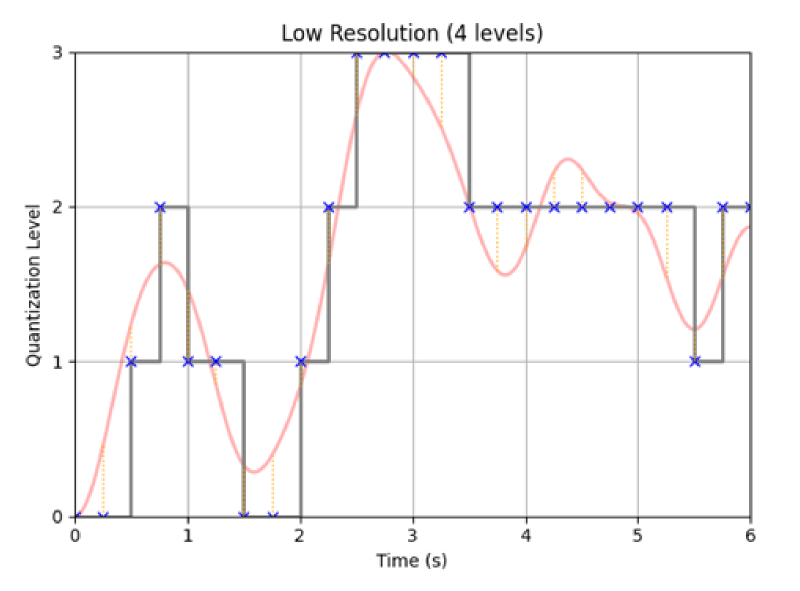


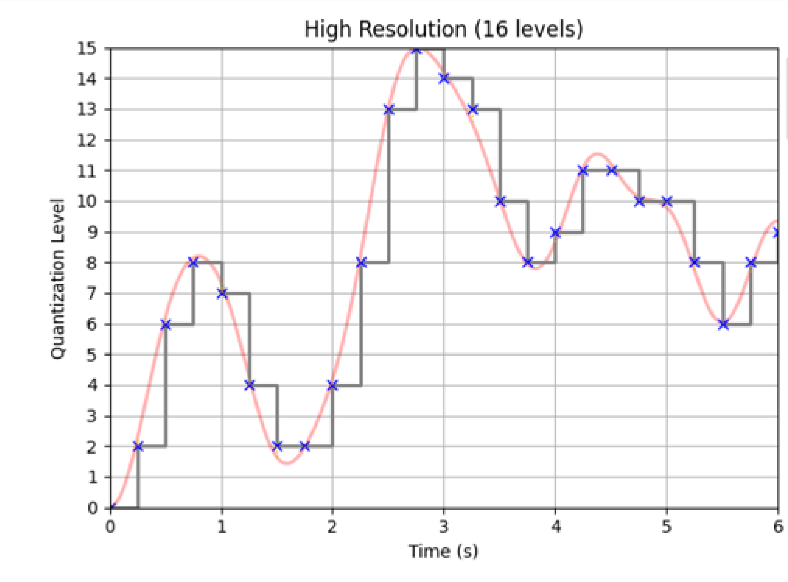
Sample Resolution

When taking a sample of the amplitude, the closest level at that time is recorded.

The higher the sample resolution the more accurate the digital representation of the sound.

The sampling using a 2-bit (4 recordable levels) sample resolution gives a less accurate digital representation then the sample using a 4-bit (16 recordable levels) sample resolution.

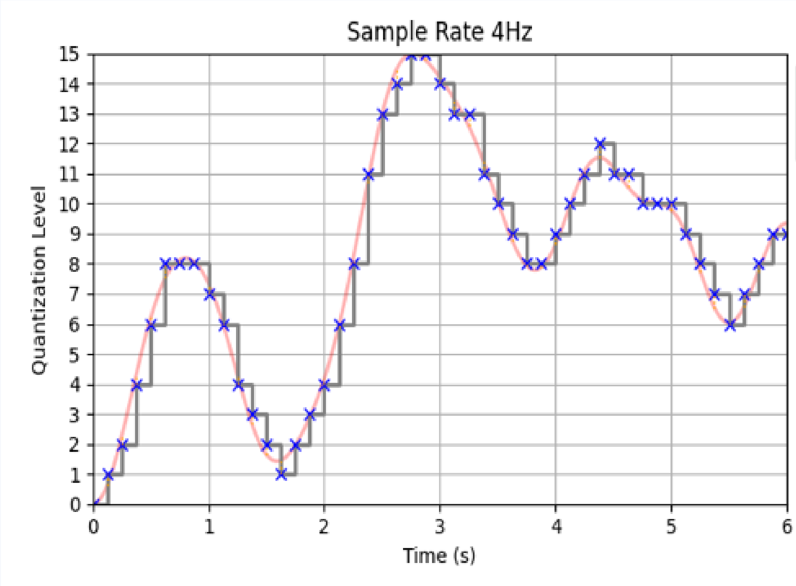


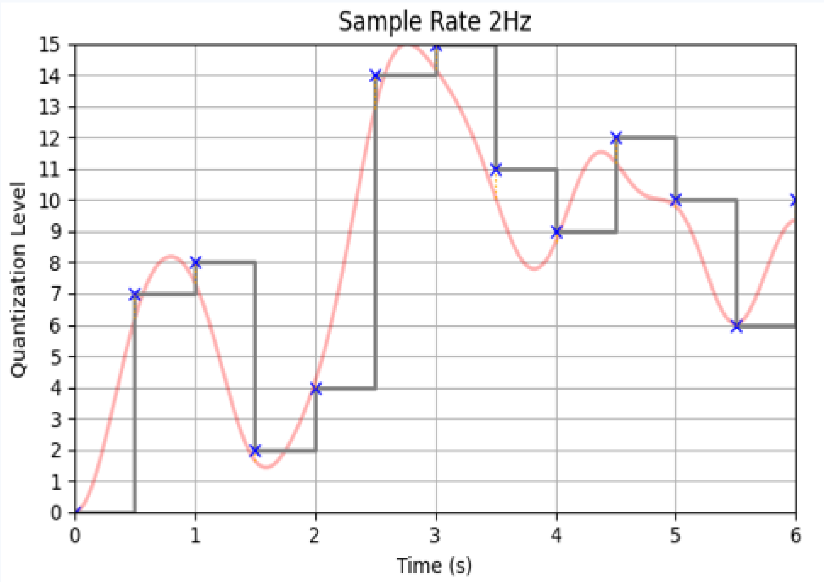


Sample Rate

The more samples taken per second the greater the accuracy of the digital representation.

The sampling using a sample rate of 2Hz gives a much less accurate digital representation than the sample using a sample rate of 8Hz.

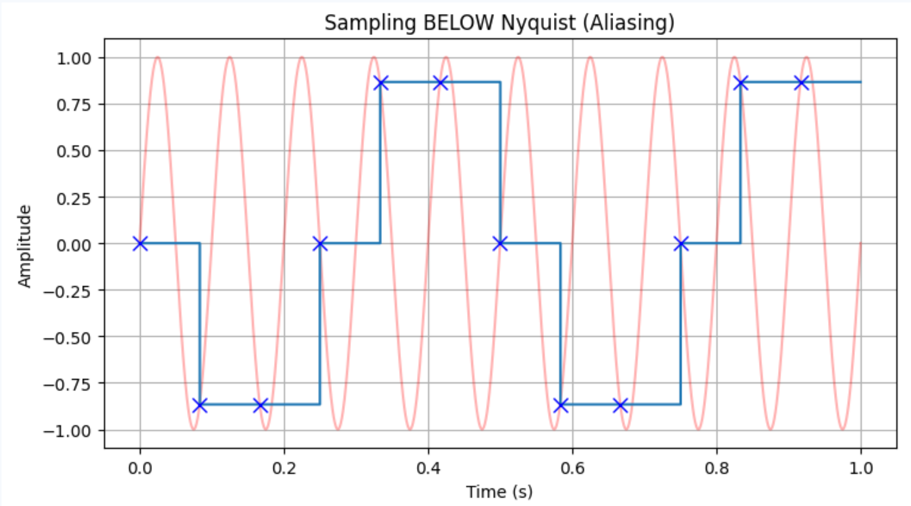




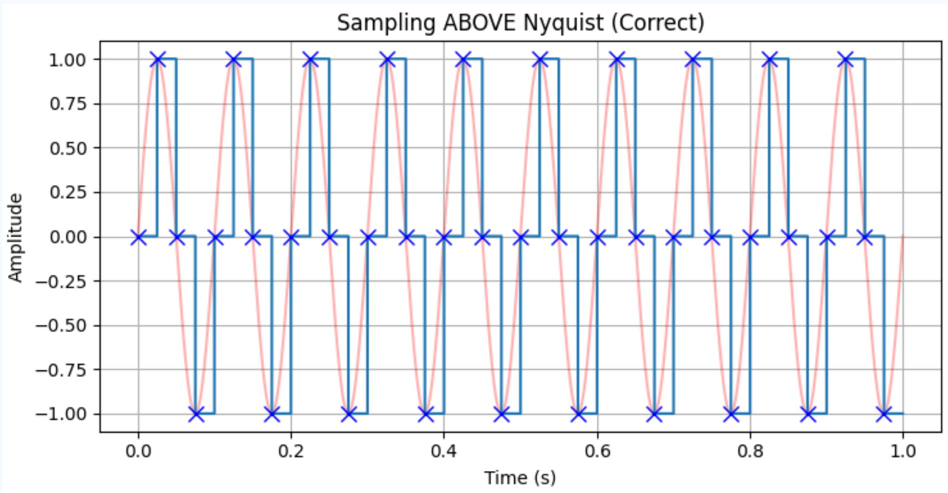
Nyquist’s Theorem

Nyquist’s theorem states that to produce an accurate recording, the sampling rate must be **at least twice** the highest frequency of the original sound signal.

In order to produce an accurate recording of a signal that has a maximum frequency of 20,000 Hz, the sampling rate would have to be at least 40,000 Hz.



- The highest frequency of this signal is 10 Hz.
- The sample rate is 12 Hz.
- The sample rate is not high enough to produce an accurate recording.



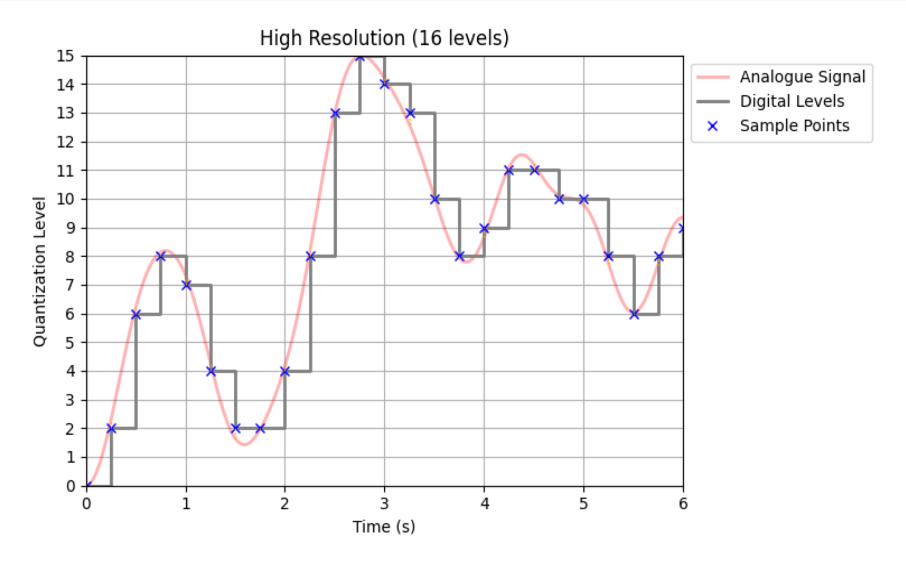
- The highest frequency of this signal is 10 Hz.
- The sample rate is 40 Hz.
- The sample rate is over double the highest frequency so the record is accurate.

Sample Storage

The **file size** of a sound sample can be calculated as:

file size = sample rate x sample resolution x length of the recording in seconds

Example: Calculating a file size for a sound sample



- This sample rate is 4 Hz
- The sample resolution is 4 bits
- The sample length is 6 seconds

The file size is 4 x 4 x 6 = 96 bits = 12 bytes

MIDI

MIDI (musical instrument digital interface) is a way of storing music as a series of event messages.

These MIDI event messages are usually 2/3 bytes long and store musical information such as:

- Turning a note on or off
- Volume
- Pitch
- Instrument
- Key pressure
- Duration

Advantages over sampled sounds

- More compact representation.
- No data quality loss through sampling.
- Easy to modify the music.
- Simple way to produce music algorithmically.
- Musical score can be generated from the MIDI file.