Real world signals are continuous. Can we represent these in digital signals?

Analogue signals can be represented as a sine wave.

For a sine wave of amplitude 1 and period T seconds:

y = sin(2pi\*t / T)

frequency is 1/T Hz

Sine waves are **continuous** in both time and value.

Sampling an analogue signal produces a **discrete** time signal, however the amplitude is still **continuous**.

**Sampling Theorem**

If the value of the signal is measured at regular points in time, **the signal becomes discrete in time, but remains continuous in amplitude.**

Sampling theorem states that we don’t need to lose anything, as long as we sample frequently enough. If the sampling frequency is more than double the maximum frequency in the original signal, nothing is lost.

To visualize this, think of how the signal is actually sampled. A measurement of amplitude is taken at a point in time, and if this measurement isn’t taken often enough, “connecting the dots” would produce a different wave.

**Quantization**

There are a limited number of bits in which we can store each sample, so we have to quantise each sample. However, quantization introduces errors in sample values.

With a large number of bits per sample, there is not much error, but storing this many bits is impractical, and too large to transmit.

When we reduce the number of bits (as in transmission), we start to notice errors, as some of the continuous values are lost. We know this as quantization noise.

Where sampling produces a discrete time signal, quantization produces discrete amplitude.

**Aliasing**

Aliasing occurs when we choose a sampling frequency of Fs, but the signal we are recording has frequencies of >Fs/2.

If these frequencies are represented as a sine wave, after sampling it becomes a sine wave of frequency Fs - F.

Fs = 8KHz, F = 5Khz

F becomes 3KHz

To visualize this, think of a high frequency sine wave, but take samples less than twice per cycle. Draw sampling points on it, and connect them, and you will get what looks like a lower frequency sine wave.

The graph representing the effects of aliasing will be a saw tooth graph.

When F = Fs, we get a strange effect due to exactly 2 samples per cycle. This produces a straight line with zero amplitude.

**Decibel Scale**

dB is a scale used for measuring the power ratio between two sounds. It is the measure of the volume of a sound, relative to the volume of another sound.

*power ratio (dB) = 10 \* log(power of my voice(W) / power of your voice(W))*

The dynamic range of human hearing is the power ratio of the loudest song we wish to hear, against the quietest sound we can hear

Dy(dB) = 10log(loudest/quietest)

This is usually about 120dB (power ratio of 10^12)

How many bits do we need to represent this?

For 120dB we needed roughly 20 bits, bit this was too much when CD’s were first invented

Settled for 16 bits, with the use of dynamic range compression.

My understanding of Dynamic Range Compression is that the whole track is turned up, and then a threshold is applied to lower the amplitude of the louder parts. Produces a narrower range of amplitudes, more consistent. This is then left for the listener to turn down at home.

**Telephone Quality Digital Speech**

Band limited to 300-3.4kHz. **Narrow band filter.**

This filter causes a loss in naturalness, but not intelligibility.

Sometimes hard to distinguish between S and F.

Sampled at 8kHz with 8 bits per sample.

64kbit/s bit-rate, but requires non-uniform quantization

**Uniform Quantization**

Each sample of speech x(t) is represented by a binary number x[n], with each binary number representing a voltage.

There is a constant difference, Δ between the voltages for adjacent binary numbers.

Delta is known as the quantization **step-size.**

However, the challenge is choosing a sensible value for Δ.

A solution to this is to adjust Δ dynamically for each sample.

**Non-Uniform Quantization**

Digitise x(t) accurately with uniform quantization, to give x[n]

Apply a ‘companding’ formula to x[n] to give y[n]

Uniformly quantise y[n] using fewer bits

Store or transmit the companding result

Can be passed through an expander to reverse the effect of companding

“It's pretty simple really. With linear quantization every increment in the sampled value corresponds to a fixed size analogue increment. E.g. an 8 bit A-D or D-A with a 0 - 1 V analogue range has 1 / 256 = 3.9 mV per bit, regardless of the actual signal amplitude.

With non-linear quantization you normally have some sort of logarithmic encoding (e.g. [µ-Law](http://en.wikipedia.org/wiki/%CE%9C-law_algorithm) or [A-law](http://en.wikipedia.org/wiki/A_law)), so that the increment for small sample values is much smaller than the increment for large sample values. Ideally the step size should be roughly proportional to the sample size. This translates to a fixed S/N ratio (due to quantization noise), regardless of the signal amplitude. Another way of looking at this is that you can use fewer bits to get a given S/N ratio over the signal amplitude range of interest.” - [**Stack Overflow on Non-Linear Quantization**](https://stackoverflow.com/questions/5321929/what-is-the-difference-between-linear-quantization-and-non-linear-quantization)

Non-Uniform quantization is done by a compander.

**Compander**

Compander increases smaller amplitudes of x[n] (reduces larger ones)

Uniform quantiser is then applied with fixed Δ.

Expander decreases smaller amplitudes of x[n] and also Δ (increases larger ones with Δ)

Produces non-uniform quantisation.

Famous companding laws are A-law & Mu-law.

Normally require 8-bits per sample.

Companding is similar to Compression, but compression is a styling technique, whereas companding is done purely for coding purposes.

**Quantization Error**

Uniform quantization produces a margin of error in each sample.

The error is random in the range (+/-)Δ/2

When sample is converted back to analogue, the error is heard as white noise.

The white noise is **spread evenly across all frequencies**. Sounds like a waterfall.

Mean Square Value of noise - (Δ^2)/12

MSV is a measure of power

**Signal to Quantization noise ratio (SQNR)**

Only applies to uniform quantization, and strictly to sine waves.

Is a measure of the “quality” of the quantization performed upon a signal.

SQNR = 10log(signal power / quantization noise power) measured in dB

A sine wave of amplitude A has an MSV of (A^2)/2

SQNR = 10log((A^2)/2 / (Δ^2)/12)

8bits per sample is not enough for uniform quantization (quieter speakers have a much lower amplitude, and hence only need 3 bits, which is too low and will generate quantization noise)

**Workshop 1 - Narrow-band speech encoding**

Over a phone, you are not listening to the full range of frequencies the human ear can detect. Lower quality.

0-4KHz can represent most of human speech. Comprehensible. Known as **Narrow Band Speech**

64kbit/s is too high to transmit via mobile phones.

Narrow Band - 13Kbit/s

Adaptive Multirate (AMR) encoding is now used. encodes Narrow Band frequencies in the range 4.75 - 12.2kbit/s

Telephone quality is achieved at 12.2kbit/s

**Differential Encoding**

Encode the difference between samples

Can be made to work at bit-rates of 32-64kbit/s

Uses lots of complex tricks, but is still not good enough for mobile phones

So far, we have tried to encode waveform shape by sampling with a limited number of bits. Uses a waveform coder.

**Parametric speech encoders**

Monitors pressure, to build a model of the person.

Vocal cords produce variance in pressure, sound. Modified by vocal tract to produce the vowel sounds (resonate).

Vocal chords held open during consonants, no vibration. Constrictions in air flow creates turbulence. White noise, random numbers used to produce excitation.

Vocal Tract model - a filter. reduces and increases certain frequencies. Produces resonation. Has some coefficients to determine which frequencies to emphasise.

Parametric Speech simply sends the following data, instead of the whole waveform:

* Fundamental Freq
* Voiced or Unv (consonant or na?)
* Gain
* Tract coefficients

**Linear Predictive speech**

Uses Parametric speech. Sender derives parameters 50 times per second, and sent to the receiver. Receiver uses these parameters to reconstruct the speech.

LPC is a form of Linear predictive speech, once used commonly in the military.

Get’s bit rate down to 2400bp/s.

Encodes 20ms speech frames, each of which consists of:

* Unvoiced/Voiced decision
* Gain/Amplitude - 8 bits
* Fundamental Frequency - 8 bits
* Leaves 37 bits for the 10 filter coefficients