Analogue to Digital and Digital to Analogue Conversion.

Implicit in Digital Signal Processing (DSP) is the ability to convert an analogue signal into its digital representation. Similarly, once all the processing is done, we must convert the signal back into analogue form so we humans can hear or see it.

Analogue to Digital Conversion (ADC or A2D)

The way this is done is to measure, or sample, the signal at regular intervals and change the amplitude, perhaps measured in volts or millivolts, into digital form.

This raises two questions:-

1. Quantisation distortion

Analogue signals are continuously variable and may adopt any level from zero to a set maximum, and may be either positive or negative. A binary word of 8 bits, eg 10010011, can have 2^8 different values, including zero. That is 256 different levels. The nearest of those 256 levels is chosen to represent the analogue signal. There will be a small error, known as the quantisation error, as a result of digitising the signal. The reason for the error is that there are only 256 different levels to choose from, rather than the infinite number needed to give an exact fit to the analogue signal. These errors can be regarded as a real signal plus quantisation 'noise', giving rise to the concept of a signal to quantisation noise ratio.

To reduce the error, more binary bits are needed. 16 bits will give 65536 levels to choose from. That is much more accurate, but at the expense of needing more bits to represent the signal. That will translate into more DSP processing power required.

Fig 1 shows part of an analogue signal and the errors in selecting the best fit digital representation. The vertical red samples are restricted to one of 8 levels each side of zero. A 4 bit binary word is being used, giving 16 values with 0000 representing the -8 level, 1000 representing the signal 0 point, and 1111 representing the +7 level.

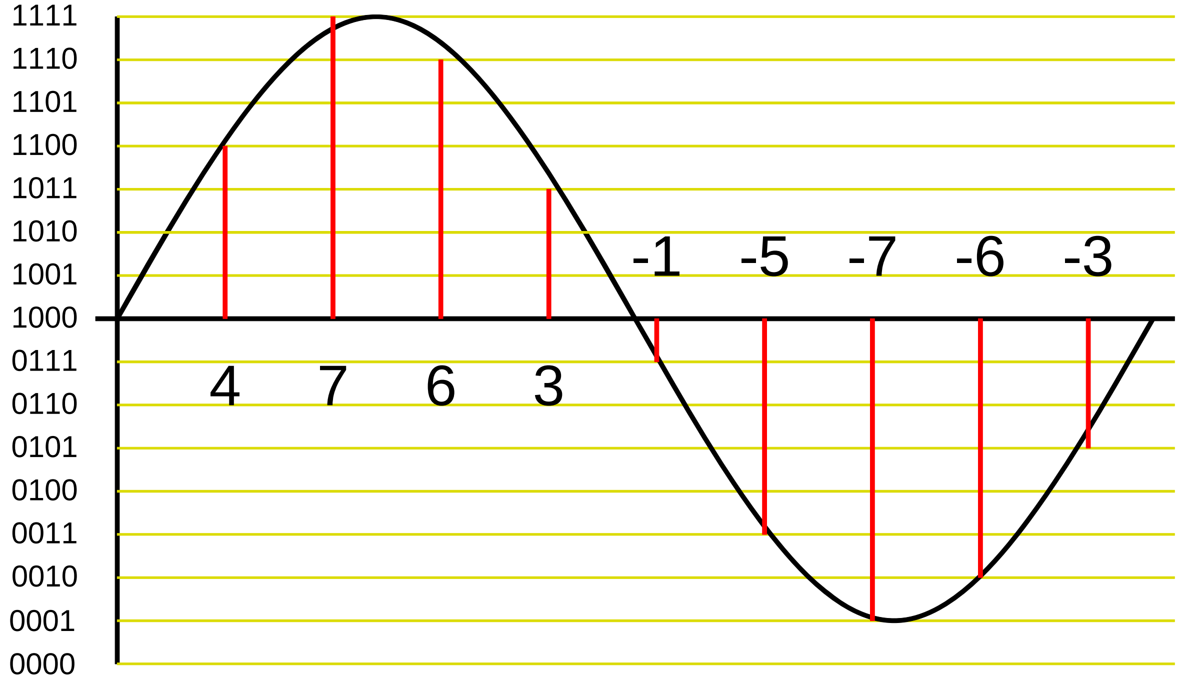

Fig 1 Digitising the amplitude results in quantisation errors

2. Sampling rate

Clearly we need to sample sufficiently often to capture all the details in the signal; but how fast is that?

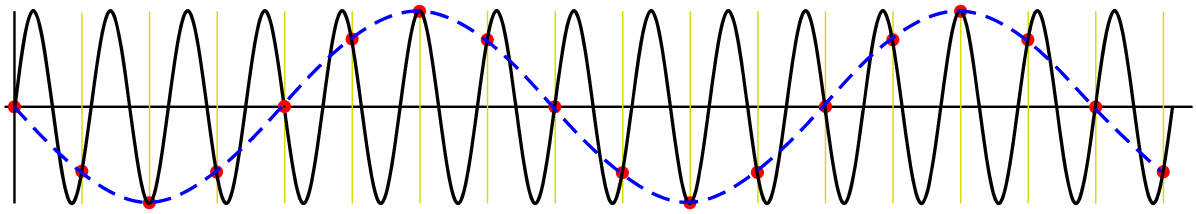

Fig 2 Under sampling the signal

Fig 2 shows a sine wave, in black, sampled 17 times in 15 cycles, that is just over once per cycle. The sample points are shown as red dots. Although the real signal is in black, the sampled points suggest the signal was the blue waveform. The sampling is too infrequent to capture all the detail in the signal.

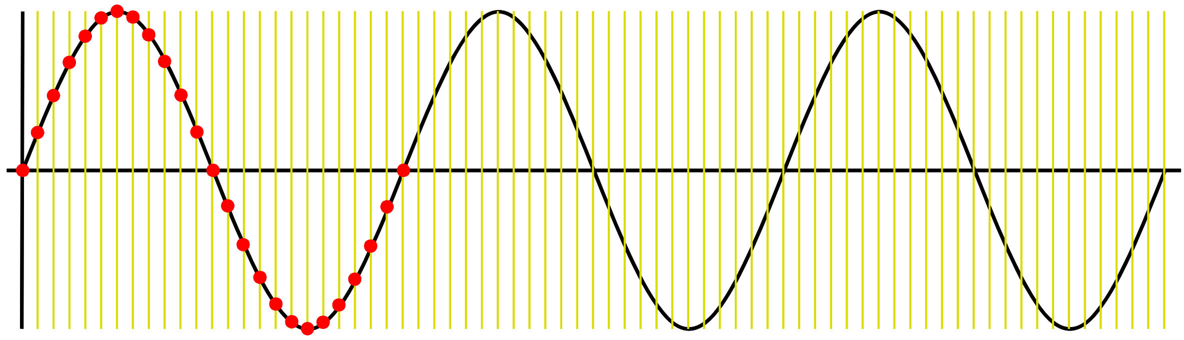

Fig 3 Over sampling the signal results in a lot of data

Clearly more samples are needed to represent the analogue signal more accurately. But just how many?

Fig 3 shows a signal being sampled very frequently. The exact shape of the signal is captured well. However fewer samples would capture this signal perfectly well enough to reproduce it properly. This results in a lot more data than needed and extra work for the digital processing system.

Finding the best sampling rate:

Fig 4 below shows10mS of time divided up into a sample taken every half millisecond, or 21 samples if we include time zero.

The blue waveform is a sine wave signal at 900Hz; there are 9 cycles in 10mS. The 0·5mS sampling points are shown by the black dots.

Also shown is an 1100Hz signal, coloured red.

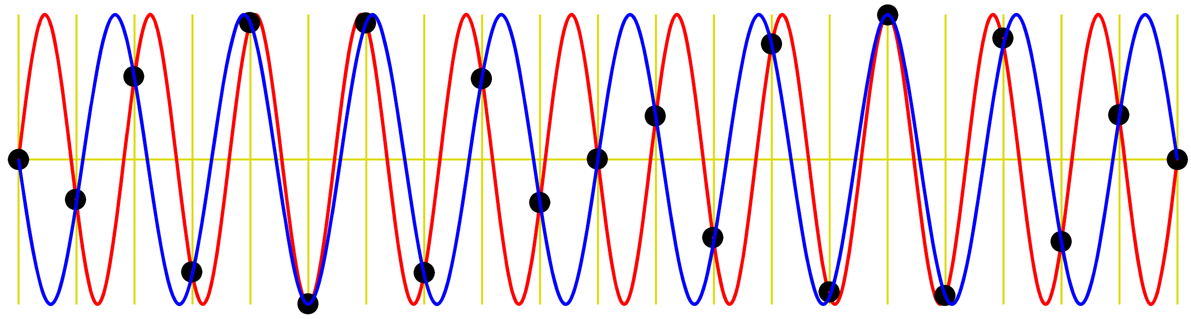

Fig 4 Sampling at the optimum rate with an unwanted higher frequency present.

It will be noticed that the results of the sampling of that signal are identical to the blue one. The sampling is failing to distinguish between a 900Hz signal and an 1100Hz one. Which one is correct?

It can be shown that it is necessary to sample at, at least twice the highest frequency signal present. The mathematics to do so are quite complex and the reader is referred to texts on the Nyquist Sampling Theorem for further information.

In the example above, it would be necessary to have a good filter in front of the analogue to digital conversion process, to remove all frequencies above 1000Hz. If there was a small trace of an 1100Hz signal allowed through the input filter, it would be processed as if it were a 900Hz signal. It is quite likely that it would interfere with the wanted signals. The 1100Hz, unwanted, signal is called an ***alias*** and the effect of these higher frequency signals appearing as lower frequency ones inside the wanted frequency range, is called ***aliasing*.** Since the input filter cannot be perfect (a good reason for having DSP!), in practice it is necessary to sample rather more often than twice the highest frequency present.

Digital to Analogue Conversion

Having carried out the necessary processing, the signal must be converted back to analogue form. There are a number of methods by which this may be done, all, now, implemented inside a single chip. A simple one to explain is the potential divider ladder.

First we must look at the principle of the potential divider.

The two resistors R are of equal value. There will be a current flowing through them due to the 4V potential difference. The PD across each resistor will be the same because the current through them is the same and the resistors are of equal value. This assumes there is no current drawn out at point A.

The PD between the zero volt line at point A will be half the 4V supply, that is 2V

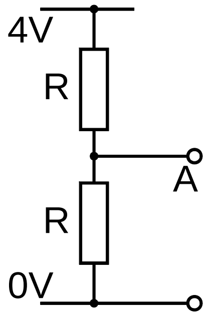

Fig 5 A Potential Divider

Now consider the circuit in Fig 6. This is also a potential divider, in fact two dividers, but drawn differently.

The 'top' resistor is the resistor R1 next to the 4V supply. The 'bottom' resistor is now made up of three resistors, R2 of twice the resistance of R1, and two resistors, R3 and R4 in series, each of value R.

R3 and R4 have a total resistance of 2R, and they are in parallel with R2 of resistance2R. The combination has an effective resistance of R, the same as the top one.

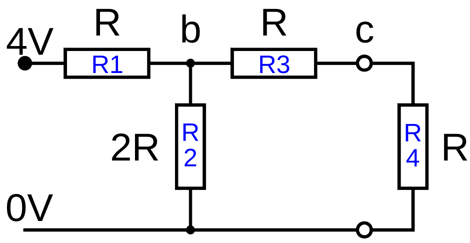

Fig 6 Two potential dividers

Point 'b' will be at half the supply voltage; that is at 2V.

Point 'c' is at the middle of a potential divider formed by R3 and R4. The PD at 'c' will be half that at 'b' since R3 and R4 are both of resistance R. The output at 'c' will be a quarter of the input or 1V.

This idea can be extended to provide outputs of 1/8, 1/16 etc, limited by the accuracy of the resistors.

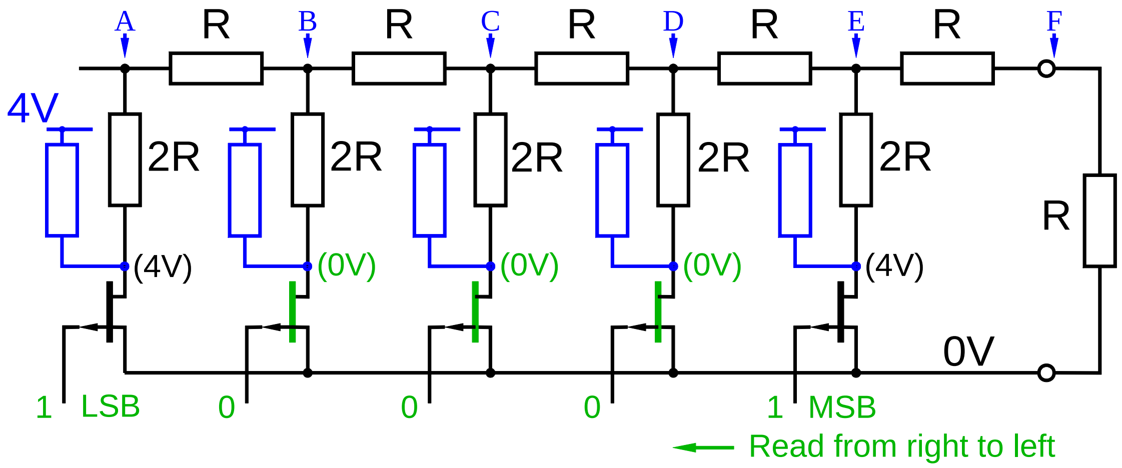

Fig 7 A 5-bit digital to analogue converter

Fig 7 shows a 5-bit digital to analogue converter based on the potential divider principle in Fig 6. The resistors to the 4V supply, shown in blue, are a lot lower in value than R and can be neglected. The bottom of the 2R resistors are connected to 4V or to 0V depending on whether the transistors are conducting (0V) or open-circuit (4V).

Point 'E' in Fig 7 corresponds to point 'b' in Fig 6. This technique is continued back up the chain or 'ladder' as it is often called. Looking to the right from points A, B, C, D and E will always 'see' 2R ohms.

If only the left-most transistor is non-conducting, point, its drain (the bottom of the 2R resistor) will be at 4V. All the others will be at 0V.

Point 'A' will be at 2V (with respect to zero), 'B' will be 1V, at 'C' it will be 0·5V, at 'D' 0·25V, at 'E' 0·125V. The output at 'F' will be 0·0625V.

If the transistors are conducting as shown in green; that is the end two are conducting, there will still be a contribution of 0·0625V from the left-most transistor, but the right-most transistor will be at 4V. On its own that would result in 2V at point 'E' and 1V at 'F'. However there is still the additional contribution on 0·0625V from the transistor at 'A'.

The output is determined by the setting of the transistors according to the table below.

Transistor					Output PD
E	D	C	B	A	
On	On	On	On	On	0V
On	On	On	On	Off	0·0625V
On	On	On	Off	On	0·125V
On	On	On	Off	Off	0·1875V
On	On	Off	On	On	0.25V
On	Off	On	On	On	0.5V
Off	On	On	On	On	1V
Off	Off	Off	Off	Off	1.9375V

The binary number controlling the transistors gives the output PD. Remember ‘On’ is a binary zero and 0V from that transistor. The binary number is also read right to left, the right-most bit is the most significant bit and the left most, the least significant bit.

Output PD = (binary value) × reference voltage (4V)/64

64 is the maximum division of this particular ladder.

Output of a DAC

Fig 8 shown the output of a DAC as the black line. It is based on the samples of the red sine wave. At each sampling point the DAC changes its output to match the present value of the red sine wave.

The DAC output is squared off into one of the 31 discrete levels with level 16 (binary 1000) being taken as the zero level allowing the signal to have both positive and negative values. Negative values are signified by binary numbers below 1000, and positive values by binary numbers above 1000.

The blue sine wave shows the average value of the DAC output.

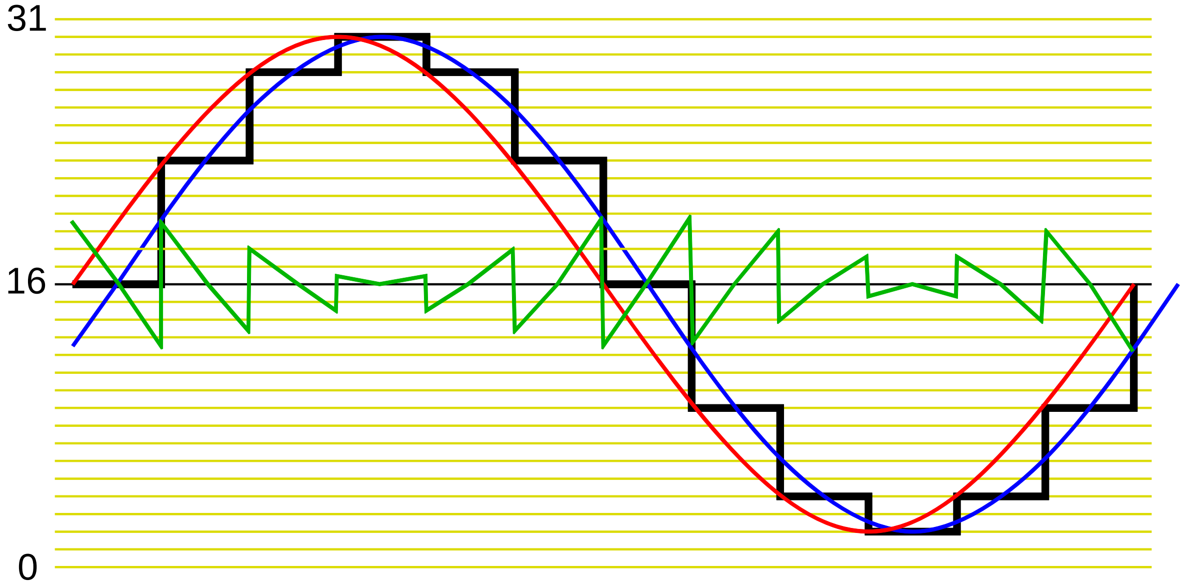

Fig 8 The DAC output of a sinewave

Two errors are noticed.

1) The output is delayed by about half a sampling period.
2) The square waveform is not a good representation of the smooth sinewave.

The timing error is of no consequence unless the original signal can be heard. On devices used to replay an earlier recording, this is irrelevant. However a digital hearing aid may result in a delayed signal to one ear, which will cause confusion to the wearer, particularly as to the direction of origin of the sound. Ideally both ears should be provided with aids, preserving relative timings.

The poor representation may be solved by filtering the output to remove higher frequencies.

Consider how the black waveform, the DAC output *differs* from the wanted output shown by the blue sine wave. Remember we are now ignoring the time delay.

Fig 9 shows a small section of the left hand side of Fig 8.

As before the black line is the actual DAC output and the blue line the part of the sine wave we really want. The green line represents the error in using the black DAC output as the sine wave.

This may be easier to visualise if the blue and green waveforms are added together. Adding the blue and green waveforms will produce the black waveform. We can say, therefore, that the green waveform represents the 'error' in regarding the black waveform as an approximation to the blue one.

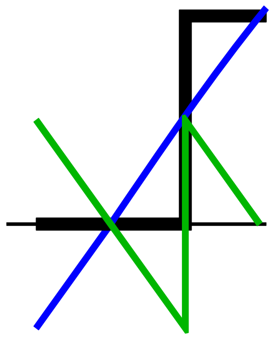

Fig 9 Section of Fig 8

In effect the actual output is the signal we really want, the blue sine wave plus the unwanted green waveform.

Inspection of the green waveform shows it is of much higher frequency. Consequently it can be removed by filtering. Passing the DAC output through a filter will remove all the higher frequency components and leave just the blue waveform. The filter has smoothed out the rough edges of the DAC output.

In practice a 5-bit DAC is too coarse for practical use. A minimum of 8 bits is required and music systems are likely to utilise 12, 14 or 16 bit DACs.

These devices are much more common than might be realised. CD and MP3 players all store the material, usually music, in digital form, which requires conversion to analogue. Mobile phones are now fully digital and require both A to D and D to A conversion.

Summary

To properly capture an analogue waveform we must sample at the rate that is more than twice the highest frequency we want to handle.

We must ensure that frequencies higher than this are not present in the analogue signal or they will appear in the captured signal as lower frequencies *inside* the wanted frequency range. This effect is called aliasing.

We must use sufficient binary bits to record the analogue signal with sufficient accuracy.

When restoring the signal back to analogue a time delay of at least half the sampling period will occur plus and delays in the digital signal processing itself.

The output is a square wave signal, which effectively contains the wanted sine wave, plus higher frequencies, which should be removed by filtering. The frequency of these unwanted signals is determined by the rate at which the signal is 'replayed'. Normally this will be the same as the sampling rate.

To simplify the filtering at the input and output it is usual to sample rather faster than twice the highest frequency.