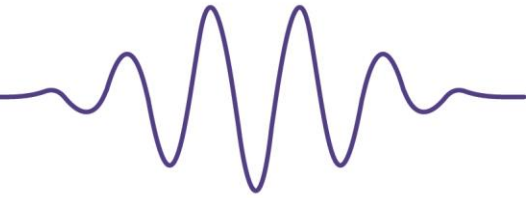
Radio Society of Great Britain

Advancing amateur radio since 1913



DIGITAL SIGNAL PROCESSING

A simplified introduction with no maths, for Syllabus 2019
tutors only, not candidates

Alan Messenger, G0TLK

Digital Signal Processing
A simplified introduction with no maths

Contents

Introduction 2
History of digital signal processing 2
Sampling 2
The basics 2
Over Sampling 2
Insufficient Sampling 3
Sampling - Nyquist limit 3
Sampling - aliasing 3
Sampling – number of bits 4
Undersampling – using Nyquist and aliasing positively 4
Sampling – restoring the analogue signal 4
Signal processing 5
Digital filters 5
A simple DSP system 5
The Fast Fourier Transform (FFT) 6
FFT analysis - windowing 6
Demonstration of a FFT and simple signal presence recognition 7
FFT – practical usage in amateur radio 7
Acknowledgements and to find out more… 8
And finally… 8

Introduction

This article, based on a presentation first given at the 2017 RSGB Convention, is intended for the amateur radio exam tutors to help with teaching the new Software Defined Radio (SDR) material in Syllabus 2019. It goes slightly beyond the syllabus requirements and is designed to give a basic background into Digital Signal Processing (DSP), enabling Tutors to answer some questions that trainees may ask, and to help tutors develop their own knowledge. Links to suggested further reading are given for those who might want to know more.

History of digital signal processing

DSP is based on mathematics and goes back quite a way:

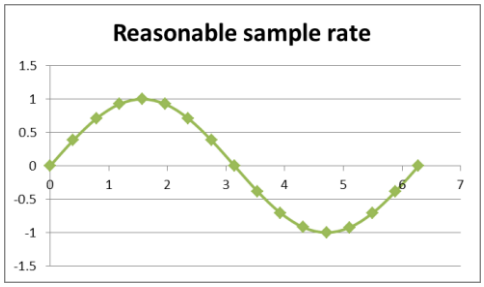
- 1779 - Laplace discovered a basic building block, the Laplace Z-Transform
- 1768 - saw the birth of Jean Baptiste Joseph Fourier who gave his name to another fundamental process – the Fourier Transform
- There it stopped for some 150 years…
- During the 1930's - the Discrete Fourier Transform or DFT was found
- 1965 - the Fast Fourier Transform (FFT) algorithm was developed from the DFT and it's this that has enabled the DSP usage we see now
- 1970's - first electronic implementations

Now - it's everywhere, for example in music files, media streaming, digital television, Wifi, mobile phones, digital cameras, image editing and amateur radio to name a few.

Sampling

The basics

Sampling is the fundamental building block for current digital signal processing systems.



Analogue signals that we can resolve with our senses need to be converted into a stream of data that can be handled by a computer – a digital representation of the analogue signal.

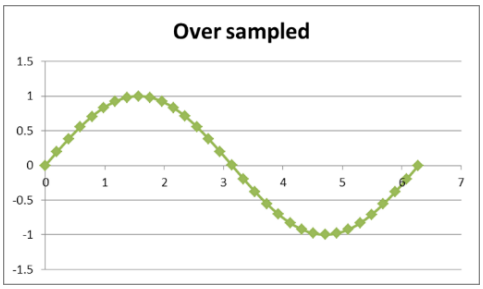
We use an analogue to digital converter, or ADC, which works by taking discrete samples at regular time intervals as shown in Figure 1.

Figure 1- Reasonable sampling

There should be just enough samples to represent the analogue signal, not too much nor too little.

Over Sampling

If we increase the number of samples the representation is more accurate but we then have to handle a lot more data in the processing of our signal.

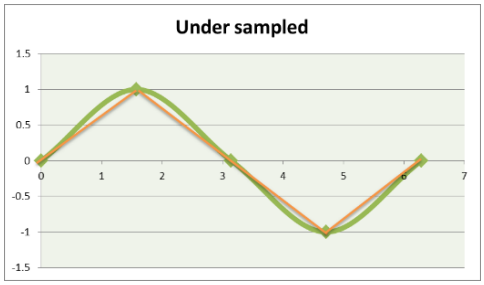


We don't want to take too many samples – there's a lot of maths in DSP and too many samples slow the calculations, which are usually time sensitive. It also means more capable (read financially expensive!) hardware.

In this example there's no need for all these sample points in order to get a representation of the input, as you can see in Figure 2.

Figure 2 - Oversampling

Insufficient Sampling



Conversely if we don't have enough samples the signal is not represented properly, shown in orange on in Figure 3.

This can happen not only in amplitude but potentially also in phase.

This gives us a classic case of "garbage in, garbage out"

Figure 3 - insufficient sampling

Sampling - Nyquist limit

It can be shown mathematically the sample rate must be more than twice the highest frequency component of the input signal. This limit is known as the Nyquist-Shannon or Nyquist limit, and if we don't satisfy it, we get trouble!

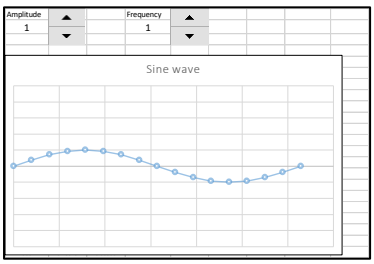


Figure 4 - Plotted sine wave

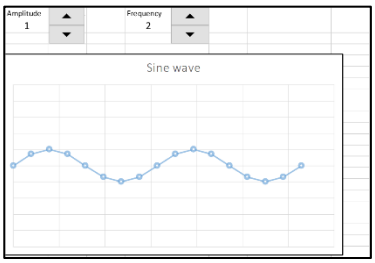


Figure 4a - Frequency doubled, still OK

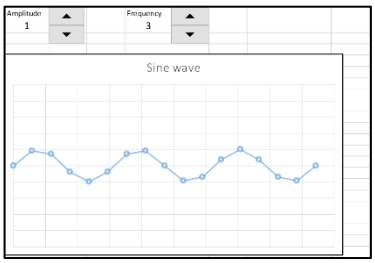


Figure 4b - Frequency tripled, losing it!

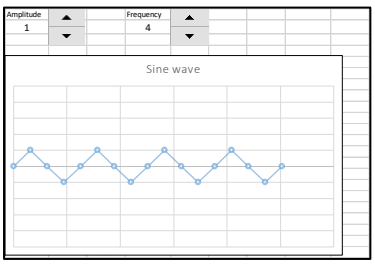


Figure 4c - Frequency changed to x4 - going!

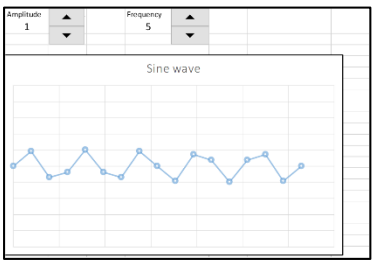


Figure 4d - Frequency is now x 5 - gone!

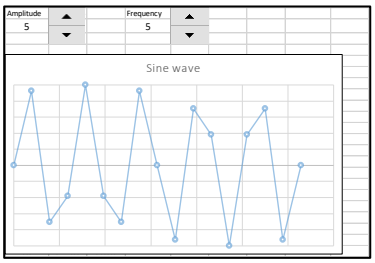


Figure 4e - Frequency x5 with larger amplitude

These calculated examples show a properly sampled sine wave, then we increase the frequency in steps, while not changing the number of samples, and see what happens when we exceed the Nyquist limit! Greater amplitude magnifies the effects.

Sampling - aliasing

With mixed frequencies on the ADC input, including harmonics, the higher frequency components can get converted into lower frequencies in a mathematical process known as aliasing, related to the ADC sample rate.

Simplistically, for a given sample rate the higher frequency components of the applied signal will exceed the Nyquist limit for that sample rate, causing unwanted products that can be likened to distortion and noise. As in analogue systems, these aliasing products are usually not wanted and can cause confusion (but see also later as in certain circumstances they can be useful).

To reduce aliasing the sample rate often needs to be considerably higher than two times the highest input frequency component, or the input must be low pass filtered. In DSP terms it's called an anti-alias filter:

Figure 5 - Anti-alias filter

Sampling – number of bits

The digital representation accuracy is also controlled by the number of bits (binary digits) in the ADC output. These control the coarseness of the digital representation, a greater number of bits meaning that there are more levels for measuring the input signal amplitude.

Most practical systems need a minimum of 8 bits although chips are available to 32 bits

- 8 bits: 0 plus a further 255 levels or, for example, 01010101 in binary notation
- 16 bits: 0 and another 65,535 levels e.g. 0101010101010101 in binary

As with a too high sample rate, if too many bits are used for a given signal then processing delay increases. There's also a cost trade-off between sample rate and bits – a 32-bit chip with maximum sample rate of 10k is about $15 but a 14-bit chip with a max sample rate of 3Gigs is $1326 (2017 prices)

You can't have a high number of bits and a high sample rate without spending serious money!

Undersampling – using Nyquist and aliasing positively

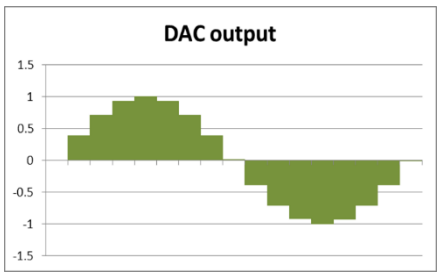
Both the Nyquist limit and aliasing are mathematically predictable and we can put that to good use. Modulation on a carrier can be recovered by sampling at the highest frequency component of the modulation envelope, but ignoring the carrier.

The carrier will be far higher in frequency than the modulation, so Nyquist will be exceeded and aliases will result. By careful design those aliases can be a copy of the modulating envelope – which is exactly what we want!

This technique uses two things that in a simple DSP system are undesirable, to produce something that is wanted. Unlike an analogue system the results are entirely predictable using maths and therefore reliable.

It's called undersampling and is a powerful widely used technique because it avoids the need for expensive high sample rate ADC's

Sampling – restoring the analogue signal



Once we have completed the signal processing the samples must be converted back to analogue form suitable for our senses.

This is carried out in a digital to analogue converter or DAC, giving a stepped representation of the signal. The sharp edges produce unwanted harmonics, so every DAC requires a suitable low pass filter on its output.

Figure 6 – DAC raw output diagram

Signal processing

Digital filters

Once we have our samples, we can process them to suit our design purposes.

Digital filters are implemented by maths instead of physical components so they can be accurately defined, are repeatable, and reliable providing the applied signals stay within the designed bounds. We can provide low pass, high pass, bandpass and virtually any other kind of filter you can think of.

Modulation and demodulation can also be carried out mathematically, and again is predictable and repeatable.

In any further reading or experiments you will almost certainly meet some terms that are important in SDR and DSP:

- I (in phase) and Q (in quadrature or 90° phase shifted) samples – are vital within SDR implementations for both modulation and demodulation
- Decimation - reducing the number of samples after processing – see https://dspguru.com/dsp/faqs/multirate/decimation/ for an explanation
- Comb filter – a series of notches in a passband
- FIR filter (finite impulse response) and IIR filter (infinite impulse response) – see https://dspguru.com/dsp/faqs/fir/basics/ for a useful comparison between the two
- Logic adders and multipliers

However, further explanation will need some maths, so is unfortunately outside the scope of this simplified introduction to DSP.

A simple DSP system

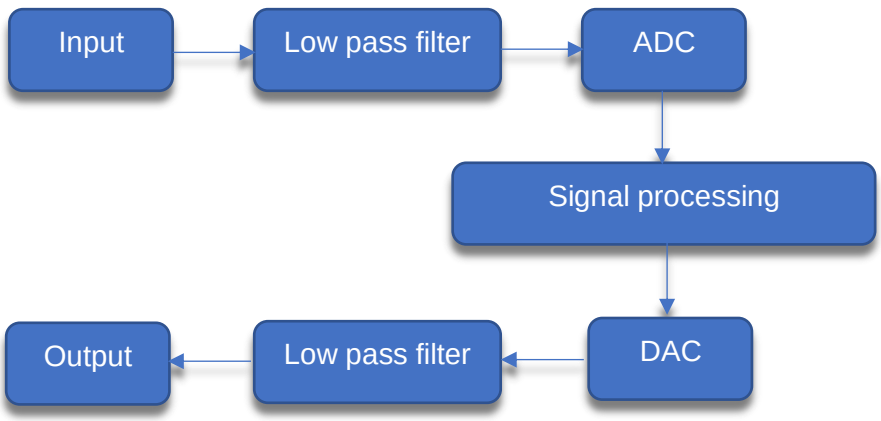


Figure 7 - simple DSP system

The above is a diagrammatic representation of what we have learned so far - a simple DSP system.

The Fast Fourier Transform (FFT)

If we want to see what's in a stream of sampled signals, we need to change them from the time domain, where we see one sample after another in the order that they were generated, into the frequency domain where we see how much of each sample there is at a given frequency.

This is used for display purposes but may also be used in signal processing. If the latter then it might be necessary to convert back to the time domain, which is done by using a FFT in reverse.

To change domains, we apply the Fast Fourier Transform. There are a number of FFT algorithms but one often met in amateur radio is known as a FFTW, which stands for Fastest Fourier Transform in the West! More details are at the project page - http://www.fftw.org/

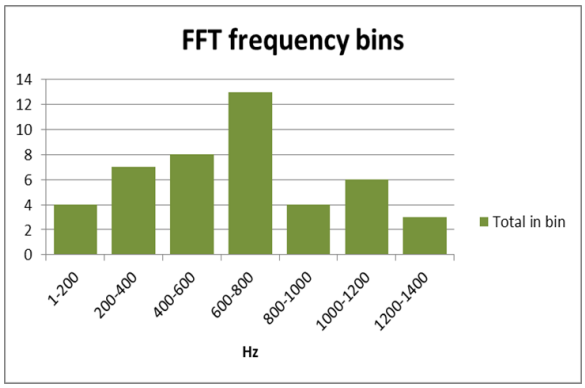


Figure 8 - FFT frequency bins diagram

A typical signal can be broken down into multiple sine waves of various frequencies.

The FFT sets up a number of "bins" for frequency ranges within that signal, which we choose at design. Very simplistically it looks in the signal for each frequency range and fills each bin to show how much of that frequency range is in the original signal.

For those who would like explore this further, from a simple explanation using a food analogy through to the maths, you could do a lot worse than https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/

FFT analysis - windowing

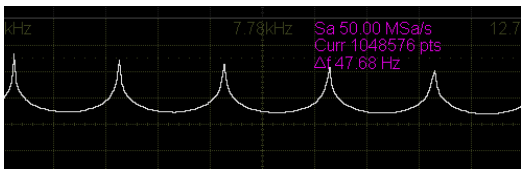


Figure 9 - Rectangular window example

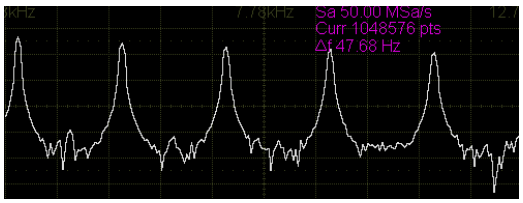


Figure 10 - Hanning window example

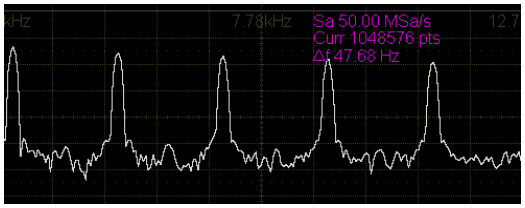


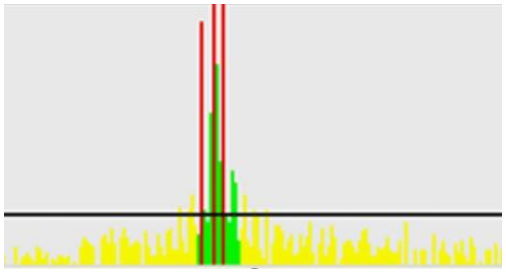
Figure 11 - Blackman window example

FFT algorithms expect a continuous stream of data, but in practice the data is finite, both within the individual samples and overall. The net result is the digital equivalent of Morse key clicks due to the sharp rise and fall times of the data stream.

This spreads out the selected frequency range into several of the FFT bins, rather than the one intended; in simple terms the bin contents overlap each other, similar to splatter in analogue signals. In DSP it's called spectral leakage, and is an effect that's definitely not wanted.

This overlap is removed or reduced by windowing maths, which smooths out the rise and fall times. There are a number of windowing algorithms, each with different results, and named after their author. It's necessary to consider which type of window best fits the data that you have. The Figures to the left are from a modern digital oscilloscope looking at a 1kHz square wave using it's in-built FFT function.

Demonstration of a FFT and simple signal presence recognition



This filter looks for a Morse signal in an audio stream:

- FFT bins in yellow
- 800Hz filter passband in green
- Red - Morse signal recognised in passband

Figure 12 - simple filter in relation to FFT bins

FFT – practical usage in amateur radio

The most common visible use of a FFT is in the waterfall and spectrum display of digital software and SDR rigs, with which we are hopefully all familiar.

- The display is built up horizontally, line by line
- Each line is divided into columns, each one of those columns representing a FFT bin, or group of bins
- The colour of the column depends on how full that particular bin is
- The whole thing repeats horizontal line by horizontal line, for example with new lines coming in at the bottom and old lines disappearing at the top as shown in Figures 13 to 15 below, or it can be the other way around (new lines in at top, old off the bottom).

When speeded up this creates the impression of vertical movement that we see on screen.

Figure 13 – time 1, start

A	C	B	E	C	A	B	D	E	C
A	C	A	C	A	B	A	C	B	C
B	A	A	C	A	E	C	B	C	D
B	E	C	A	A	E	D	E	C	D

Figure 14 – time 2, new line changes

A	C	B	E	C	A	B	D	E	C

Top line moves off screen

A	C	A	C	A	B	A	C	B	C
B	A	A	C	A	E	C	B	C	D
B	E	C	A	A	E	D	E	C	D

New line comes in at the bottom

E	C	D	B	B	C	C	D	B	A

Figure 15 – time 3, revised display

A	C	A	C	A	B	A	C	B	C
B	A	A	C	A	E	C	B	C	D
B	E	C	A	A	E	D	E	C	D
E	C	D	B	B	C	C	D	B	A
and so on (it can be reversed so the display moves downwards)									

Acknowledgements and to find out more…

Thanks to the following whose resources have been valuable to my learning about DSP and in preparation of this article:

Reading material:

- Generally - Wikipedia - https://en.wikipedia.org
- RSGB Radio Communications Handbook – including the use of Microsoft Excel™ to chart and demonstrate sampling, IQ signals and filters; the book is to be found at - https://www.rsgbshop.org/acatalog/Online_Catalogue_Technical_6.html
- DSPGuru - more advanced topics including more on filters - https://dspguru.com
- Q&A - others will have had the same questions as you! - Signal Processing Stack Exchange - https://dsp.stackexchange.com
- FFT explanation - https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/

Experimenting and learning with live DSP:

- Audacity software - http://www.audacityteam.org - DSP on live audio signals
- SDR usage - WSTX-X – found at https://physics.princeton.edu/pulsar/K1JT/wsjtx.html - and HDSDR software - www.hdsdr.de - although there are of course many others – see the RSGB book Software Defined Radio to be found at - https://www.rsgbshop.org/acatalog/Online_Catalogue_Computing___Radio_39.html#a1627 - for example, plus various proprietary implementations.

There's plenty of information out there, some of it quite heavy and mathematical, but hopefully this lightweight introduction has been useful.

And finally…

Our quick skate through DSP has been described in one dimension. To scramble your brains a bit more consider that modern digital bandpass filters, for example, are often calculated in two dimensions…

We are certainly not going to go into modern multidimensional digital signal processing if for no other reason than it's completely beyond this author!

Good luck with your teaching.

Alan Messenger, G0TLK April 2019

Sampling charts produced using Excel, which is a Microsoft Corporation registered trademark.