Introduction

Contents

1 Digital Signal Processing

1.1 What is Digital Signal Processing?

Digital Signal Processing (DSP) is a set of techniques that typically include such things as:

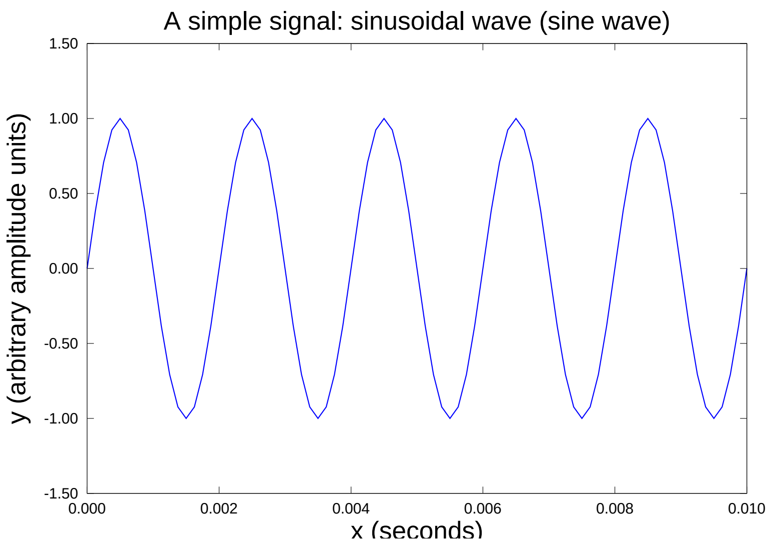
- Filtering
- Frequency domain techniques (*i.e.* Fourier)
- Time domain techniques
- Random signals
- Predication and Estimation (*e.g.* time series estimation)

DSP has many applications including:

- Audio processing
- Communication systems
- Image processing
- Video processing
- Data compression
- Vehicle control
- Financial engineering

1.2 What is a Signal?

DSP is a about processing of signals that are typically in digital form. Signals are therefore very important to DSP. A simple example of a signal is a sine wave, e.g.

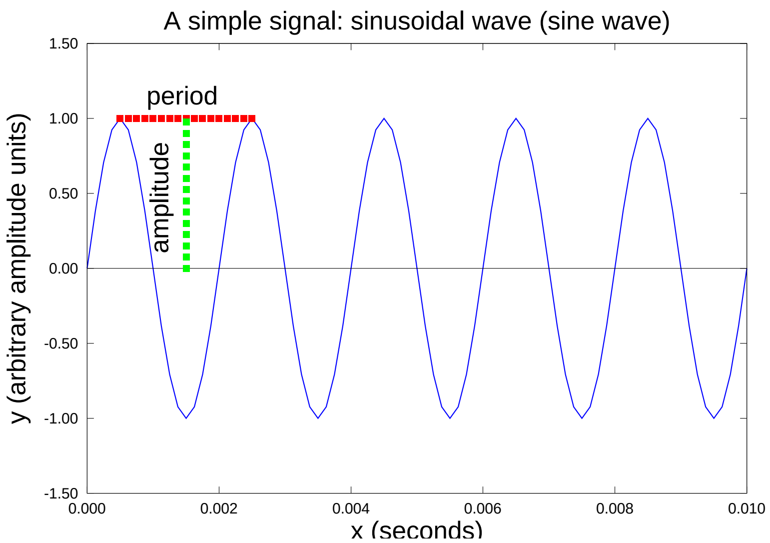


A signal can contain information for many reasons including:

- *Communication*
- *Storage*
- *Calculation*

Information is carried in the signal typically by varying one or more of the following parameters:

- *amplitude, "a"*;
- *period, "T"*;
- *frequency, "$f = 1/T$"*;
- and *phase, "ϕ"*.



We should all be familiar with the equation for a sine wave:

$$y(x) = a\sin(2\pi f x + \phi)$$

where "x" is time in seconds for this example. Amplitude "$a = 1$" controls the height of the wave.

Frequency and Period are very important. In particular, the topic of frequency is studied in great detail in DSP:

- f is the frequency
- Measured in Hertz or Hz
- Here period, $T = 0.002$s
- $f = 1/T$ Hz, therefore $f = 1/0.002 = 500$Hz.

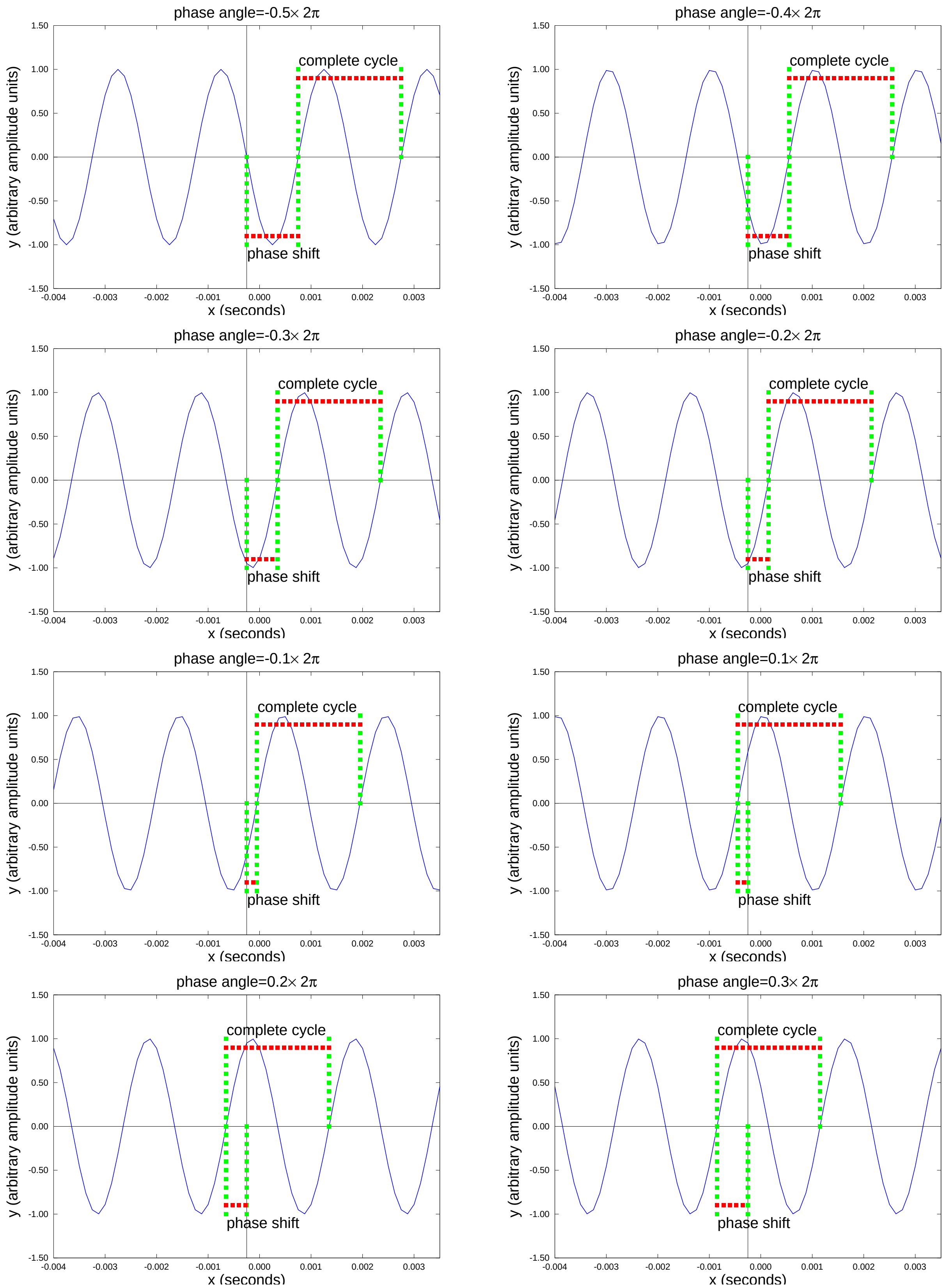
Phase Phase is important as it indicates the signal has been delayed for some reason. This may be used to encode important information, e.g. in Phase Shift Keying.

- ϕ is the phase
- Here $\phi = 0$

For the above example, $\phi = 0$, so

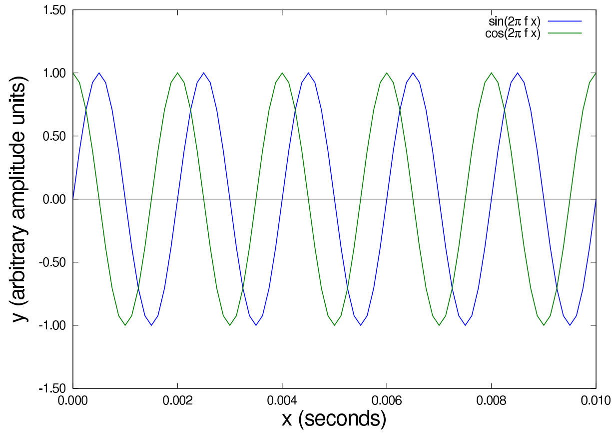
$$y(x) = y(x, \phi = 0) = a \sin(2\pi f x).$$

Examples of non-zero phase can be seen in the following:



1.3 Cosine Vs Sine

Cosine and Sine functions are equivalent except for a phase shift ($1/4\times$period).

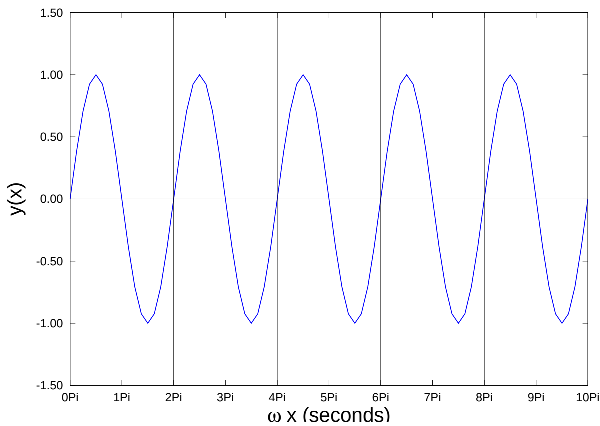


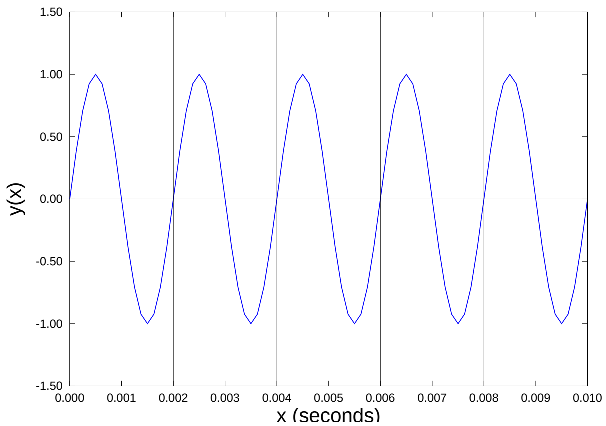
- $\cos(2\pi fx) = \sin(2\pi fx + \phi)$ where $\phi = \pi/2$.
- $\sin(2\pi fx) = \cos(2\pi fx + \phi)$ where $\phi = -\pi/2$.

1.4 Angular Frequency

- Frequency, $f = 1/T$
- Angular frequency, $\omega = 2\pi f$
- 1 period or cycle $= 2\pi$ radians

$$\begin{aligned} y(x) &= \sin(2\pi fx + \phi) \\ &= \sin(\omega x + \phi) \end{aligned}$$





1.5 Phasor Representation

As already mentioned, signals such as sine waves have frequency associated with them as an inherent property, especially for signals that *repeat* or which are periodic. In any case, the concept of frequency is something that we will be looking at in great detail in this course using techniques such as the Fourier transform. A Fourier transform is a mathematical operation that converts the signal to a representation that helps us to understand the frequency content of a signal. We will not just yet look at the Fourier transform. However we do need to know about complex numbers as the Fourier transform is defined in terms of complex numbers. So let us remind ourselves about complex numbers. Recall that a cosine (or sine) wave:

$$y(x) = a\cos(\omega x + \phi)$$

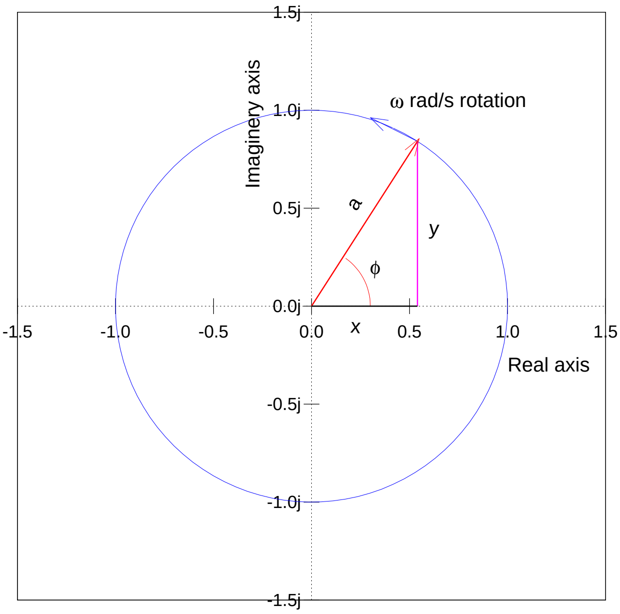
can be represented as a phasor. A phasor is a complex number:

$$z = x + jy = a(\cos(\phi) + j\sin(\phi))$$

where x is known as the real part or $\mathrm{Re}(z) = x$ and y is known as the imaginery part or $\mathrm{Im}(z) = y$. x and y can be calculated with $x = a\cos(\phi)$ and $y = a\sin(\phi)$.

Also remember $j = \sqrt{-1}$.

Argand *or* Phasor Diagram:



1.6 Complex Numbers

The square root of minus one is not defined so a symbol, j is used (sometimes i):

$$j = \sqrt{-1}.$$

Powers of imaginary numbers have the following properties:

- $j^2 = -1$
- $j^3 = -j$
- $j^{-1} = \frac{1}{j} = -j$

If $z = x + jy$ (rectangular form) then alternative representations are:

- Polar form: $z = a\angle\phi$
- Exponential form: $z = a\exp(j\phi)$

where $a = \sqrt{x^2 + y^2}$ and $\phi = \tan^{-1}(y/x)$.

1.7 Properties of Complex Numbers

If $z = x + jy$, $z_1 = x_1 + jy_1$ and $z_2 = x_2 + jy_2$ then

- Addition: $z_1 + z_2 = x_1 + x_2 + j(y_1 + y_2)$
- Subtraction: $z_1 - z_2 = x_1 - x_2 + j(y_1 - y_2)$
- Multiplication: $z_1 z_2 = a_1 a_2\angle(\phi_1 + \phi_2)$
- Division: $z_1/z_2 = a_1/a_2\angle(\phi_1 - \phi_2)$
- Reciprocal: $1/z = 1/a\angle(-\phi)$

- Square root: $\sqrt{z} = \sqrt{a}\angle(\phi/2)$
- Complex conjugate: $z^* = x - jy = a\angle - \phi$

The polar form simplifies some operations such as multiplication and division of complex numbers.

Euler's identity is an important identity in DSP because it tells us that a complex exponential is actually equivalent to a real cosine signal combined with an imaginary sine part.

$$\exp(j\phi) = \cos(\phi) + j\sin(\phi)$$

The complex exponential is used in the Fourier transform and in many other aspects of DSP. Using Euler's identity we can state the following:

- $\cos(\phi) = \mathrm{Re}(\exp(j\phi)) \longrightarrow$ or the real part, x
- $\sin(\phi) = \mathrm{Im}(\exp(j\phi)) \longrightarrow$ or the imaginary part, y

Recall the cosine wave:

$$y(x) = a\cos(\omega x + \phi)$$

which can be written as:

$$y(x) = \mathrm{Re}(a\exp(j(\omega x + \phi))) = \mathrm{Re}(a\exp(j\omega x)\exp(j\phi))$$
$$= \mathrm{Re}(A\exp(j\omega x))$$

where A is the phasor representation of $y(x)$ given by

$$A = a\exp(j\phi) = a\angle(\phi).$$

1.8 Complex Exponentials, Sines and Cosines

The complex exponential has the following properties:

- $y_1(x) = b\exp(j\omega x) = b\cos(\omega x) + jb\sin(\omega x)$
- $y_2(x) = b\exp(-j\omega x) = b\cos(\omega x) + jb\sin(-\omega x)$
  as
  - ◇ $\cos(-\omega x) = \cos(\omega x)$ (even function)
  - ◇ $\sin(-\omega x) = -\sin(\omega x)$ (odd function)

This means that

$$y_1(x) + y_2(x) = 2b\cos(\omega x).$$

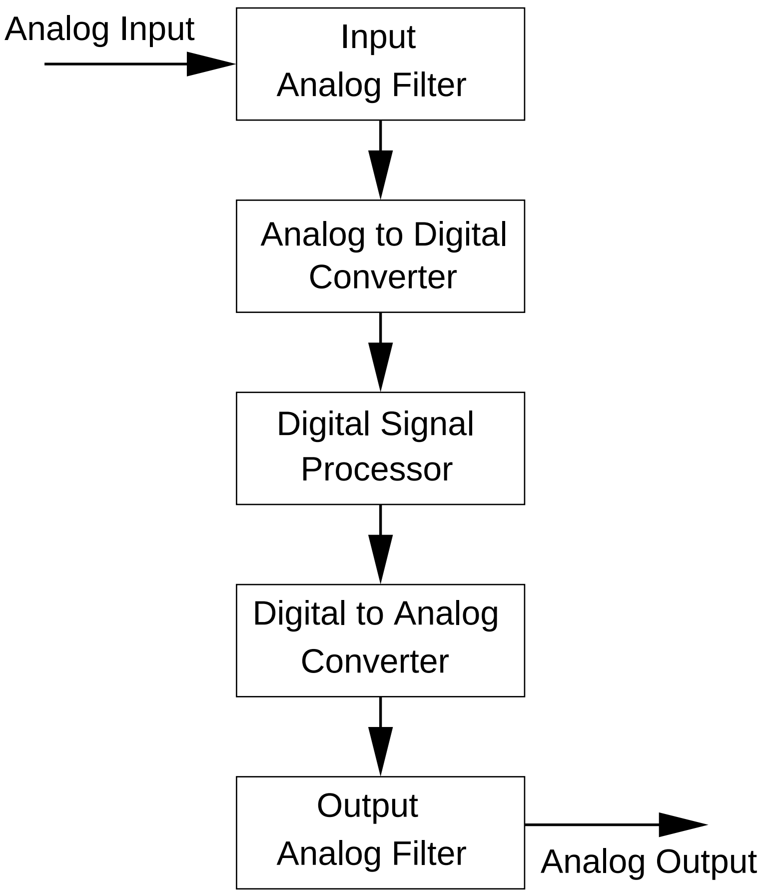
So that

$$b\cos(\omega x) = \frac{b}{2}\exp(j\omega x) + \frac{b}{2}\exp(-j\omega x).$$

A similar approach can be used to derive a sine function.

2 A Digial Signal Processing System

A typical DSP system consists of a number of stages, often including an input i n the form of an analogue signal and perhaps even an output in the form of processed analog signal. In between the steps to represent the signal in the digital form will include a Digital Signal Processor that performs some sort of DSP function:



In summary the steps illustrated above perform the following operations:

- Input Analog Filter (antialiasing): *Limits frequency range*;
- Analog to Digital Converter: *Converts signal to digital samples*;
- Digital Signal Processor: *Storage, Communication and or Calculations*;
- Digital to Analog Converter: *Convert to continuous signal*;
- Output Analog Filter: *Removes sharp transitions*.

2.1 Analog to Digital Converter

The analog to digital converter is an important component in many DSP systems and even many other digital systems that may not be immediately considered to be a DSP type system.

- Real world is typically *analog* (continuous) however computers are digital and if we want to perform some sort of computer operation on a signal we therefore need to convert it to digital form.
- Digital signal approximates analog signal with discrete quantised samples
- ADC converts an analog signal to a digital signal
- Signal is digitised in two ways:
  - ◇ Signal is sampled at a sampling rate or frequency: Information is collected about the signal at regular intervals.
  - ◇ The continuous or analog signal is then quantised: *i.e.* put into digital form, where only a finite set of numbers are represented.

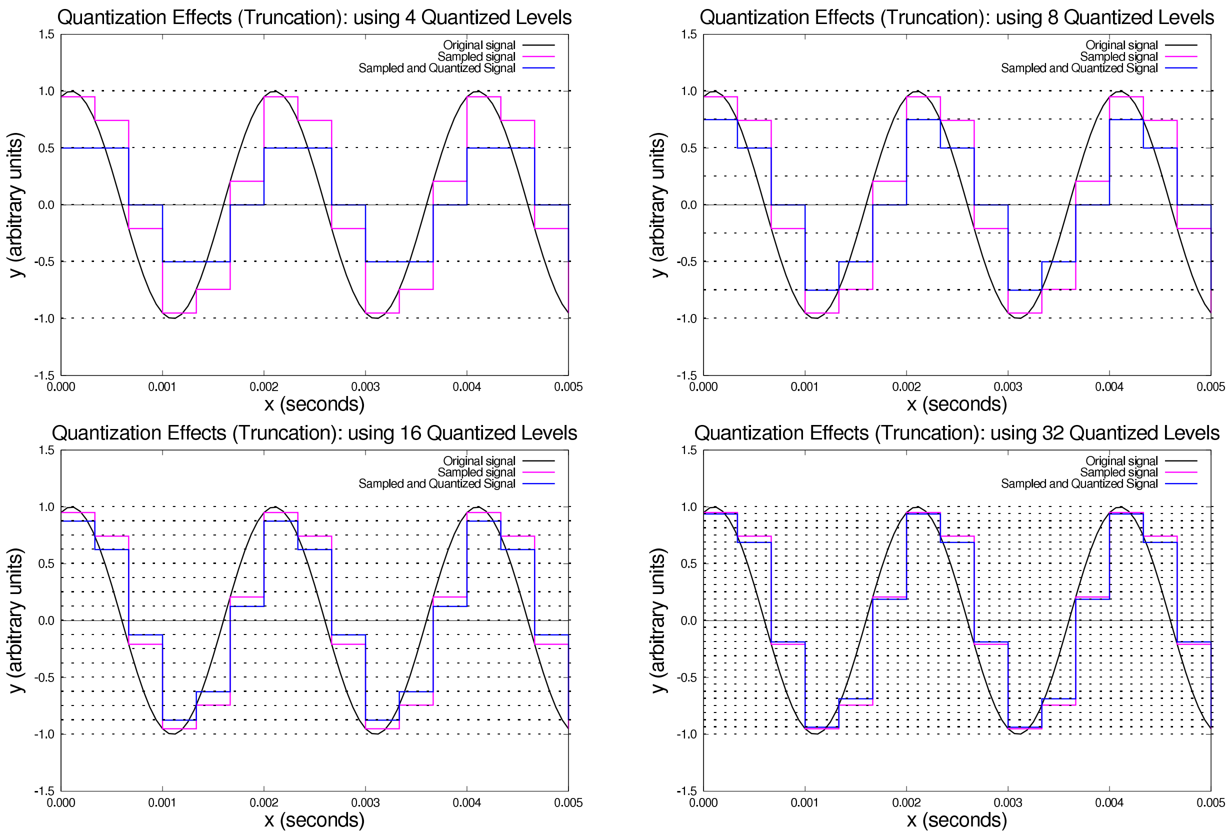
2.2 Quantisation using Truncation

Unfortunately the digitisation or ADC does remove some important information. Part of the digitisation process is known as quantisation which means:

- Signal can be quantised using *e.g.* truncation where numbers following specified position are removed.
- Examples:
  - ◇ 5.7 truncated to integer is 5
  - ◇ 5.11 truncated to 1 decimal place is 5.1
- Negative numbers are truncated in the same way (note different to the common *floor* function in matlab), *e.g.*
  - ◇ -5.78 truncated to integer is -5
  - ◇ -5.135 truncated to 2 decimal places is -5.13

2.3 Truncation Quantisation *examples*

It is important to understand the implications of quantisation. Illustrations of various levels of quantisation can be seen in the following:



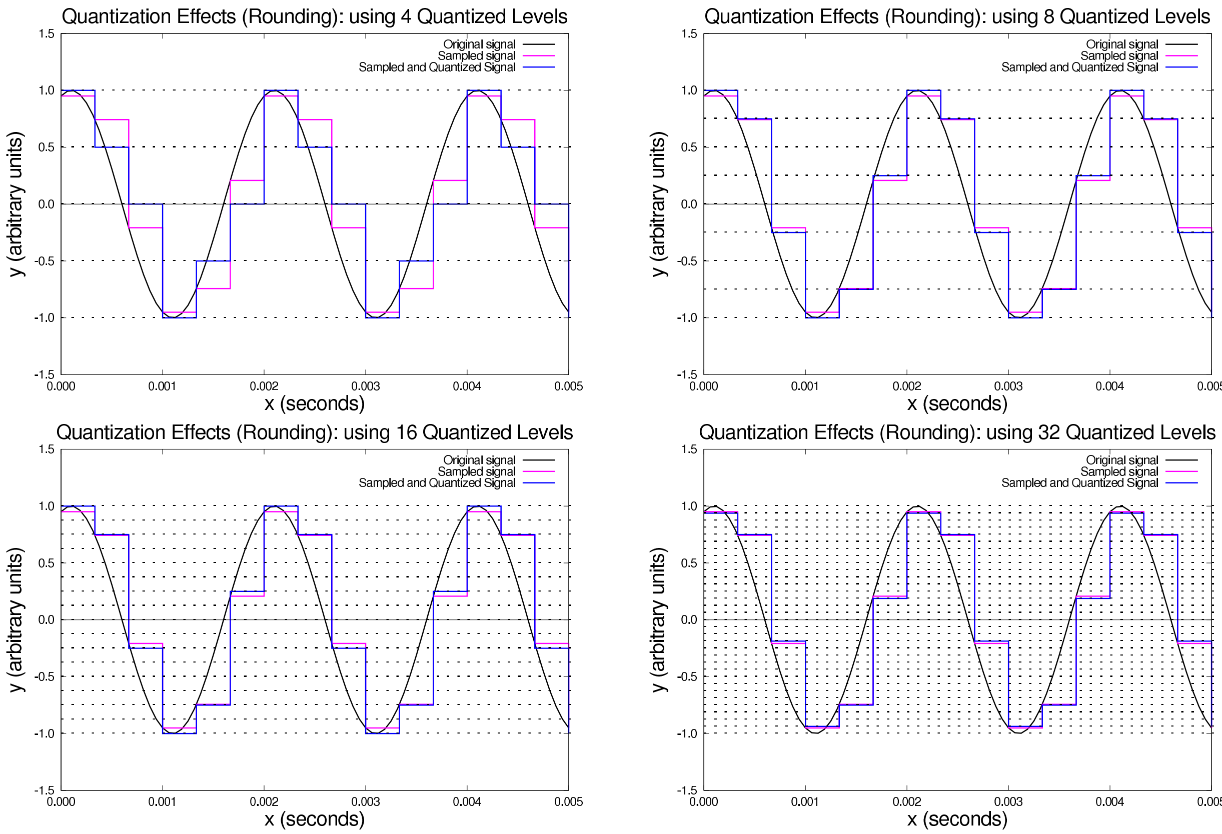
- Errors can be seen between the sampled and the sampled and quantized signals.

These errors are typically referred to as quantisation noise.

2.4 Quantisation using Rounding

- Rounding can be a quantization method associated with smaller errors, *e.g.*
  - ◇ 5.7 rounded to nearest integer is 6
  - ◇ 5.11 rounded to 1 decimal place is 5.1
  - ◇ -5.78 rounded to nearest integer is -6
  - ◇ -5.135 rounded to 2 decimal places is -5.14

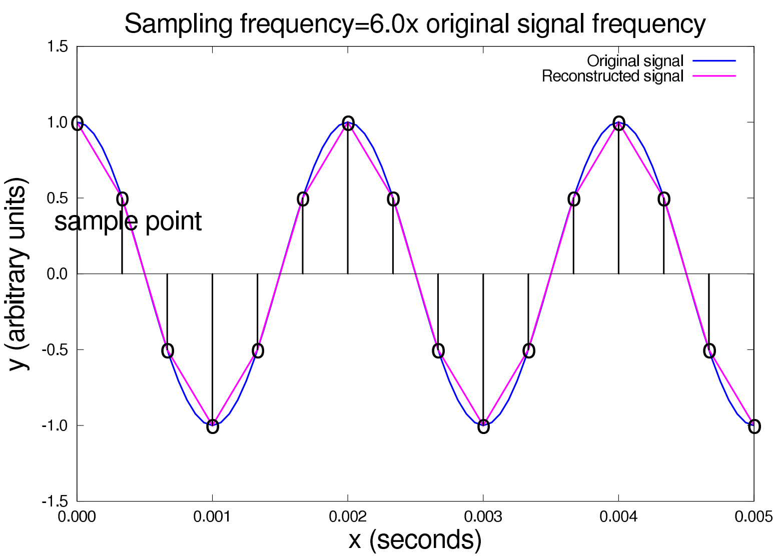
2.5 Rounding Quantisation *examples*



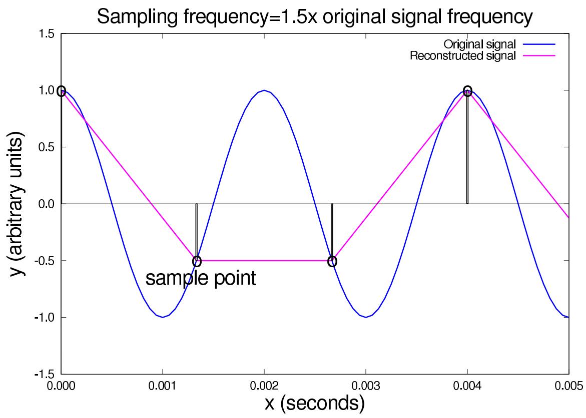
- Errors can be seen between the sampled and the sampled and quantized signals.

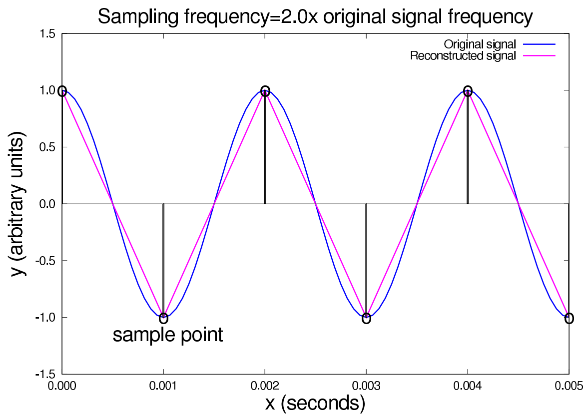
2.6 Sampling

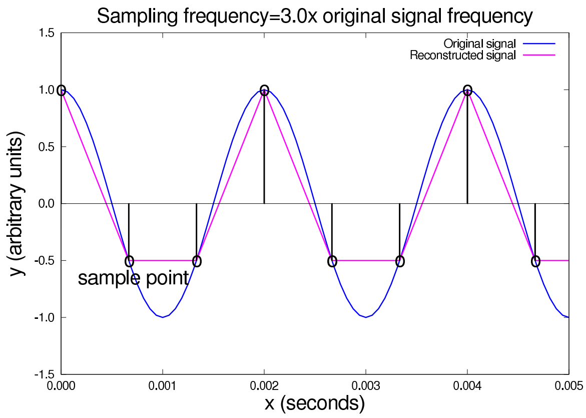
- Sampling also affects the quality of the digitised signal.
- Higher sampling rate reduces error and enables better representation of the original analog signal in digital form.

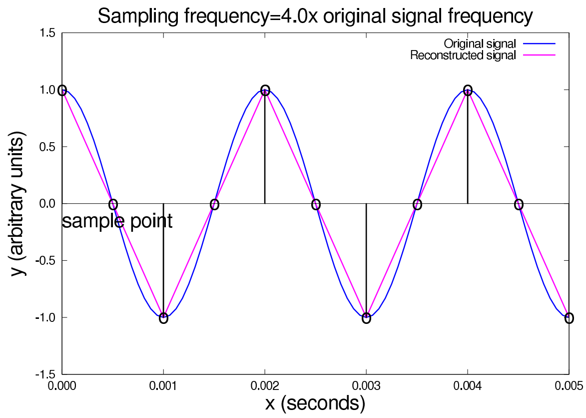


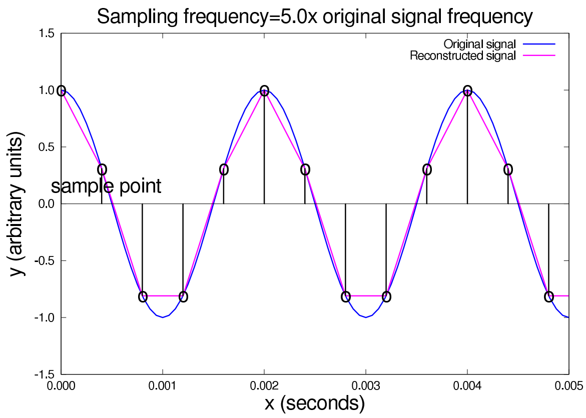
2.7 Sampling *examples*

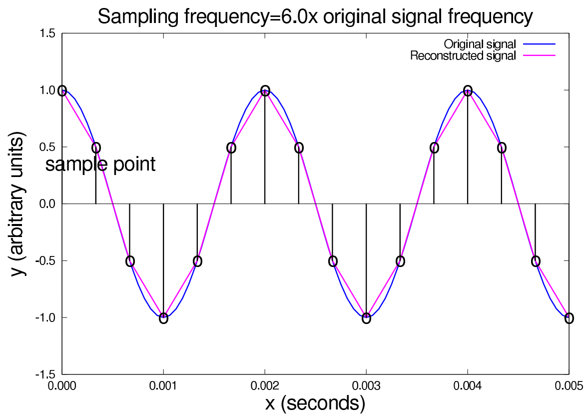


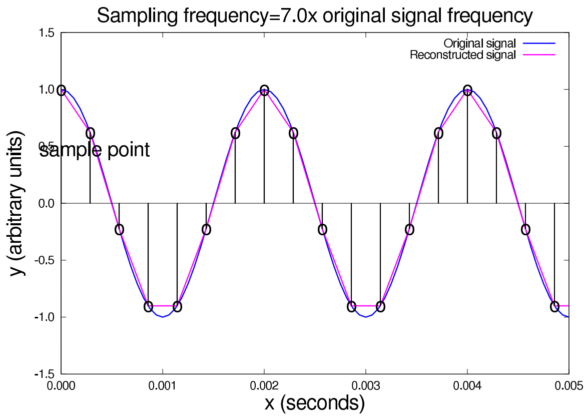


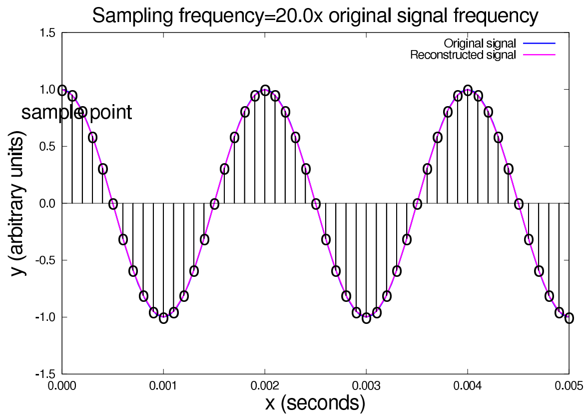






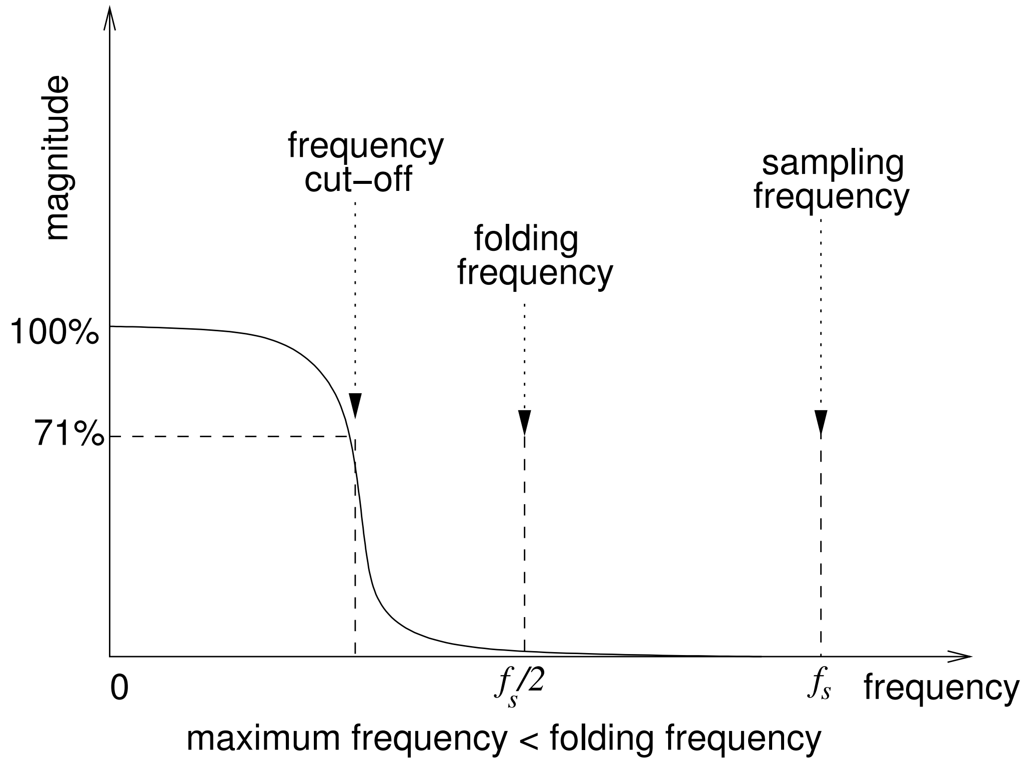






2.8 Input Analog Filter: Antialiasing Filter

- Analog to Digital Converter (ADC) requires signal below a particular frequency known as the folding or Nyquist Frequency[1]
- $\therefore$ Limit frequency range to below Nyquist frequency ($f_s/2$) before Analog to Digital Conversion. The antialiasing filter needs to remove all signal above the Nyquist frequency otherwise aliasing will occur where the inadvertently high frequency signal content will not be sampled at a sufficiently high enough frequency and aliasing frequencies will occur in the digitised signal.

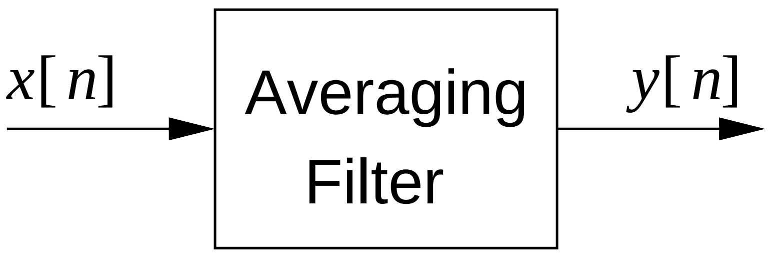


- Otherwise next stage produces frequency errors (*i.e.* aliasing)
- Sampling produces copies of signal at multiples of sampling frequency
- Aliasing occurs when copies of signal overlap each other

Aliasing frequencies will also be present in the signal even after conversion back to analog form. So all frequencies have to be removed above the folding frequency. This means that the cut-off frequency of the antialiasing filter needs to be below the folding frequency because there will still be signal content above the cut-off point. The cut-off frequency is the frequency at which the filter is removing the signal content by half-power (or 71% magnitude). This means there will still be signal content above the cut-off frequency. So the cut-off frequency has typically to be much less than half the sampling frequency.

2.9 Digital Signal Processor

After digitisation (with the ADC) digital signal processing may then be performed on the digitised signal. One of the simplest examples of a DSP technique is the process of averaging over a number of digitized samples.



[1] Nyquist was a scientist working at Bell Labs in the 20th century. His first name was Harry.

Consider a signal, x that is indexed by a discrete variable n:

$$x[n]$$

we can look back in time by subtracting from the time index n, e.g.

$$x[n-1]$$

will tell us the value of the signal for a previous time instance. We can look further back in time by subtracting higher numbers, e.g.

$$x[n-2]$$

will tell us the value of the signal two times instances previously. Remember that the signal is always changing as time goes by, which means that n is always increasing with time. We can therefore average over the current signal value along with some previous signal values, whatever values they might take. This can be expressed very simply with:

- Averaging filter:

$$y[n] = \frac{x[n] + x[n-1] + ... + x[n-k+1]}{k}$$

  for window width $k = 3$

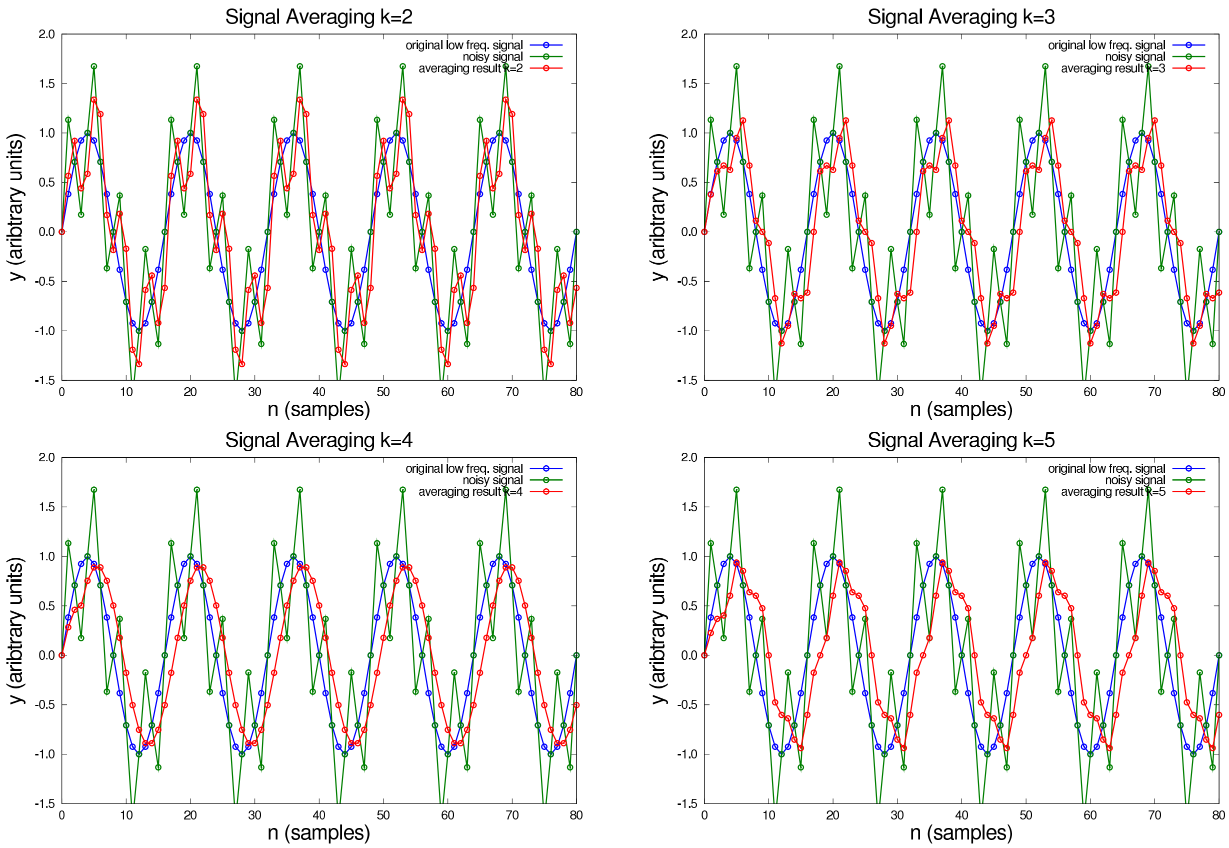
$$y[n] = \frac{x[n] + x[n-1] + x[n-2]}{3}$$

  where $x[n]$ is an input value at sample time n and $y[n]$ is an output at sample time n

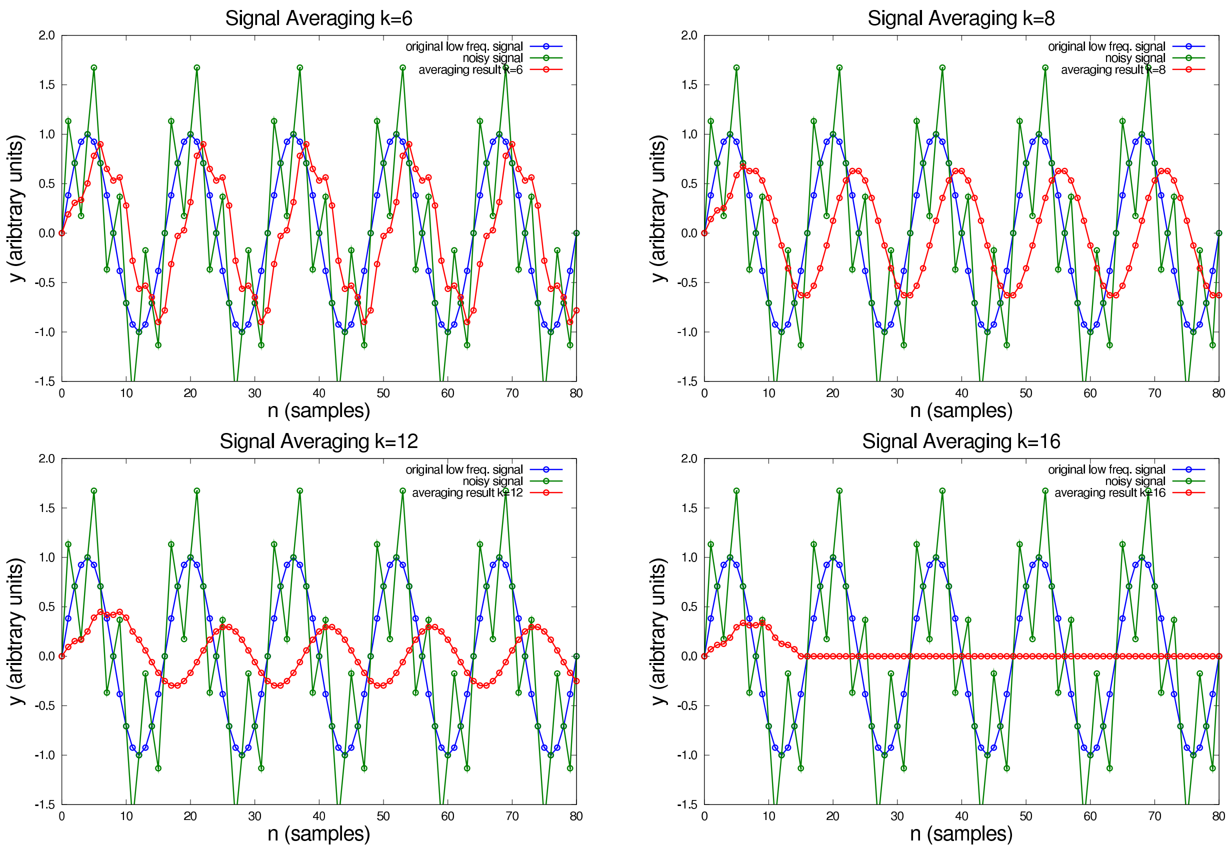
The output of the averaging filter y will be a smoother version of the input signal x where sudden changes have been averaged out. Some examples of applying this simple averaging signal to an example signal can be seen following.

2.10 Averaging Filter *Examples*

Window width k controls the response of the filter. If k is too low, there is little benefit on output signal.



Window width k controls the response of the filter. If k is too high, the filter removes all of the output signal.



3 Summary

This has just been an introduction to DSP. There is much more to understand and DSP is a rich set of techniques that can be applied to any type of signal to help understand the frequency content, to improve the quality of a signal or more.