Research Centre for Non-Destructive Evaluation

Engineering Doctorate Programme

Introduction to Signal Processing for NDE

Dr Steve Sharples (convenor)
and
Prof Richard Challis FREng (author)

Optics and Photonics Group
Faculty of Engineering
University of Nottingham

RCNDE
UK Research Centre in NDE

Revision history:
July 2008, revision 04.
June 2009, revision 05.
June 2010, revision 06
June 2011, revision 07
June 2012, revision 08
July 2013, revision 09
June 2015, revision 10
June 2016, revision 11
File: EngD_DSP_Course_Notes

Introduction to Signal Processing

Table of Contents.

Introduction to Signal Processing

List of Figures.

Introduction to Signal Processing

0. Introduction, by Richard Challis.

This introductory course on signal processing has been designed specifically for Engineering Doctorate (EngD) students working in the field of non-destructive evaluation (NDE). It is expected that the day-to-day scientific and technical activities of these students will require many different approaches to signal processing and the interpretation of numerical data more generally. They will have come from a variety of academic backgrounds and will therefore have very different experiences of signal processing – some will have covered the subject over a full four year MEng programme whilst others will only have touched the surface of a few basic concepts such as correlation and spectrum analysis. I have designed the course so that it will enable beginners to rapidly develop understanding of basic concepts and to get started on practical assignments. I very much hope that the approach that I have taken will also provide new insights and stimulation for those students who have already experienced signal processing at some depth.

In preparation for the course I have had brief discussions with most of the academic staff of RCNDE in order to appraise myself of the many perceptions of what aspects of signal processing are most important for NDE. Also of importance were the requirements for an understanding of signal and system theory to support the course on ultrasonic transducers to be given by Strathclyde University.

Much of the foregoing would imply that the type of course that is required for the EngD programme would need to contain most of the signal theory part of a full MEng in Electronic Engineering and, clearly, such coverage will not be possible within the confines of a limited five-day teaching exercise. Instead, I have selected a few important aspects from all four years of a typical MEng, and have arranged the material in a different order from that of a typical EEE course. For example, correlation precedes Fourier analysis, and many ideas about discrete (sampled) signals precede the treatment of continuous time signals and systems using Laplace and Fourier transforms. Formal coverage of sampling theory comes after much work on discrete data operations. Some material on analogue systems is included since these remain important in many experimental systems – their inclusion is the reason that the name of the course is *Signal Processing* as distinct from *Digital Signal Processing*. Digital filters are introduced relatively late in the course, followed closely by discussion of the z-transform.

An important aspect of the course will be supervised computational exercises which will both illustrate key aspects of the course and also provide for the development of theoretical understanding and practical skill. The course will consist of five one-day sessions, each of which will contain two, three or four nominally one-hour lectures, followed by exercise classes.

Finally – these notes may include some linguistic and algebraic errors, as well as imperfections in some of the figures, for all of which the author apologises in advance.

1. Types of Signal.

Analogue signals, also known as *continuous time signals*, are simply quantities that are defined continuously in time and, mostly, vary as a function of time. Examples are the continuous voltage from a microphone, or an ultrasonic transducer.

Discrete signals, also known as *digital or sampled signals* are usually versions of analogue signals that have been sampled at equal instants in time so that the original signal can be stored on a computer as a series of numbers. The process of forming the discrete signal from the analogue one is known as *digitisation* or *analogue to digital conversion*. The device that achieves the operation is an *analogue to digital converter* (ADC); conversion in the opposite direction is known as *digital to analogue conversion*, the device being a *digital to analogue converter* (DAC). A discrete signal is usually characterised by a *sampling frequency* and a *resolution*, or *bit depth.*

Sampling frequency is the number of samples of the signal that are taken per unit time during the digitisation operation; the units are samples per second, often expressed as Hz, samples s^{-1}, or Ss^{-1}. The Nyquist sampling therorem states that **A continuous signal with frequency components in the range f=0 to f_{max} Hz can be reconstructed from a sequence of equally spaced samples, provided that the frequency of sampling, exceeds 2 f_{max} Hz.**

Note: f_{max} must include **all** of the components in the signal – the useful information as well as the parts of the signal that have no use, such as noise. If the signal is not sampled at an adequately high rate then the resulting discrete version will be unusable; here the underlying phenomenon is called a*liasing* because higher frequency components appear in the discrete version *alias* low frequency components. Think what an undersampled audio signal would sound like.

Bit depth is the number of binary bits used to store the discrete signal, typically greater than eight and less than 64. Currently we are using ADCs which sample in the tens of MHz range with 14 bit resolution.

Binary signals are pulse waveforms which exist in only two states, 0 and 1; a square wave is the simplest example.

Point process signals simply record the time at which an event occurs, such as crosses marked on a time axis. They are not generally associated with an amplitude, but they can be – in which case they would form a discrete signal with the amplitudes at all ‘non event’ times set to zero.

2. Classes of Signal.

All of the above signals can fall into one of the following classes, which are not mutually exclusive.

Periodic – such as a fixed frequency sine wave which is continuous over all time.

Aperiodic – the signal has no observable periods but may not be random. These are sometimes known as *energy signals*, such as the sound of a guitar string plucked once.

Deterministic signals have structure in that some linear analysis on part of the signal may be used to predict or *determine* the future course of the signal.

Random signals have no *structure*, although they do have *properties*. They are indeterministic in that the data could have occurred in any order and the signal would not appear to be different.

Stochastic signals combine both deterministic and random components. Most real engineering signals fall into this class because we have the 'useful' components of the signal with unwanted electronic noise superimposed. There is a decomposition theorem which applies to many signals which states that they can be broken down into a pair of signals which are uncorrelated to each other and in which one is wholly deterministic and one is wholly random.

3. Signal Properties in the Time Domain.

Randomness. We may wish to answer the question 'is the signal random'. There are many tests for randomness which are beyond the scope of this course, although we will re-address the question later when we cover *autocorrelation*.

The **signal mean** is the value of a signal averaged over time. In complex signals it can be taken to mean the 'dc' or zero frequency component. It is expressed algebraically

as $$\bar{x} = \frac{1}{T}\int_0^T x(t)dt \qquad \text{(continuous)} \qquad (1a)$$

or $$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x(n) \qquad \text{(discrete)} \qquad (1b)$$

The **signal variance** expresses its variation about its mean value, it generally applies to random (noise like) signals and expresses their *power*. For a discrete signal

$$\mathrm{var}(x) = s^2 = \frac{1}{N-1}\sum_{n=1}^{N}\left[x(n) - \bar{x}\right]^2 \qquad (2)$$

s is the standard deviation and expresses the variation of the signal in amplitude units, such as volts.

The **mean-square** (MS) value of a signal expresses its power. It is related to variance, but includes the mean (dc) component.

$$x_{MS} = \frac{1}{N}\sum_{n=1}^{N} x^2(n) \qquad (3)$$

The **root mean-square** (RMS) value of a signal expresses its effective amplitude.

$$x_{RMS} = \sqrt{x_{MS}} \tag{4}$$

Signal to noise ratio (SNR) expresses the relative amplitude of a signal and any associated noise, usually in units of dB. Imagine observing a noisy sine wave on an oscilloscope – the waveform will appear 'fuzzy' due to the noise. The computational problem is to separately quantify the signal and the noise so that SNR can be estimated – it is not always easy to do this and approximations may be necessary – it does not matter much what those approximations are, provided that (i) the estimates are consistent throughout a measurement programme, and (ii) that a clear statement is made as to the assumptions and the mechanism of the approximation. The problem is often fairly easy to address in the context of NDT where we generally encounter signals in the form of short pulses. The part of the waveform which contains the pulse is selected by window in the time interval T_1 to T_2, $\Delta T_{12} = T_2 - T_1$, and its mean square value x_{RMS} is calculated. The same operation is then done on a part of the waveform which is thought to contain noise alone, over interval T_3 to T_4, $\Delta T_{34} = T_4 - T_3$.
The SNR then becomes

$$SNR = 20\log_{10}\frac{\sqrt{\dfrac{1}{\Delta T_{12}}\sum_{T_1}^{T_2} x^2(t)}}{\sqrt{\dfrac{1}{\Delta T_{34}}\sum_{T_3}^{T_4} n^2(t)}} \quad dB \tag{5a}$$

or

$$SNR = 10\log_{10}\frac{\dfrac{1}{\Delta T_{12}}\sum_{T_1}^{T_2} x^2(t)}{\dfrac{1}{\Delta T_{34}}\sum_{T_3}^{T_4} n^2(t)} \quad dB \tag{5b}$$

3.1 Simple Measures to Improve SNR.

Hardware. Careful design of electronics and wiring interconnects will minimise noise for a given hardware configuration.

Filtering. Very often the noise superimposed on a signal will cover a wider frequency range than the signal proper. It is therefore possible to apply a filter which passes the signal and rejects part of the noise. Clearly there will be some frequencies which are common to both the signal and the noise and these noise components will remain present. The filters used are generally lowpass, but in specific cases they can be designed to reject specific bands of frequencies, such as 50Hz/60Hz mains interference and/or its harmonics. Once a signal has been digitised the filters can be applied numerically. However, it may be necessary to apply an analogue filter to a signal before it is digitised to reduce the chance of *aliasing* in the digitisation process, and this filter can improve SNR as well.

The coherent average. In many experiments a repeating timing signal or other stimulus is used to obtain an ensemble of records of notionally identical responses which differ from each other only by virtue of the random noise which is superimposed on each record. The idea of the coherent average is that these records are lined up in the time domain, one below the other, so that they can be added together in a manner which synchronises their time origins:

$$\bar{x}(t) = \frac{1}{M}\sum_{m=1}^{M} x_m(t) \qquad (6)$$

The resulting SNR in $\bar{x}(t)$ will be $\sqrt{M}$ times better than the original, so for 100 averages the SNR improves by factor 10, for 10,000 averages the factor is 100. This is the coherent average; in signal processing operations all over the world it is probably done tens of thousands of times each day! Applications include EEG processing in psychology, video picture improvement, ultrasonic, optical and electromagnetic instruments, see figure 1.

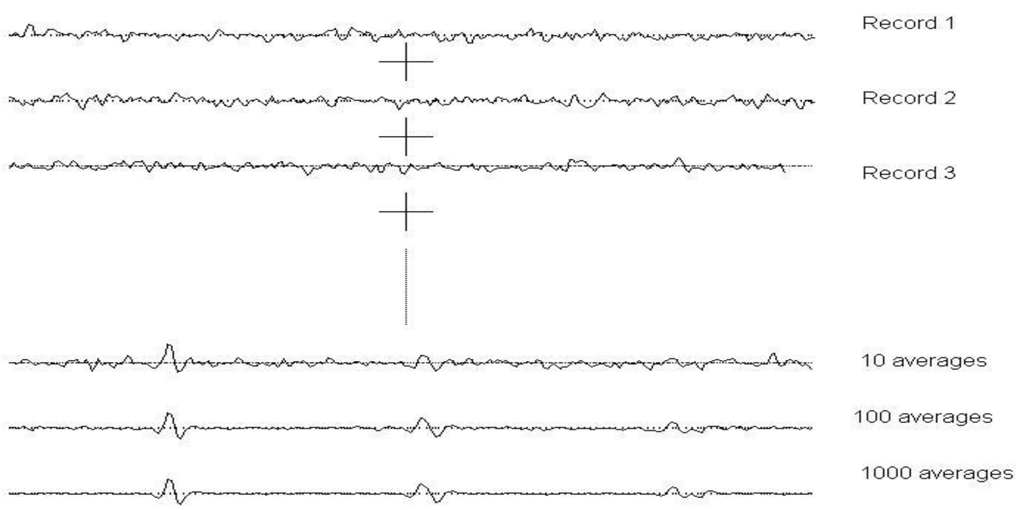


Figure 1. Example of a coherent average: Ultrasound propagation in a highly absorbent medium.

3.2 Signal Trends.

There are situations in which the properties of a signal change with time, and if this change has identifiable structure then it is known as a *trend*. In a general sense trends are often difficult to identify and each case needs to be addressed individually. However, there is a fairly simple trend which is often found in experimental signals and this is the *base line shift*. Basically the signal we wish to study is an oscillation about a baseline value which has to be removed so as not to affect calculations on the signal proper. If the base line shift is a simple dc offset then we can calculate the mean of the signal and subtract it so as to leave the ‘wanted’ part of the signal. However if the offset changes with the time course of the signal we need to recognise this change, quantify it and extract it from the signal. If the base line shift is linear with time we can fit a straight line and assume that this represents the trend; the trend could be higher order, but it would be dangerous to try to fit above order 2.

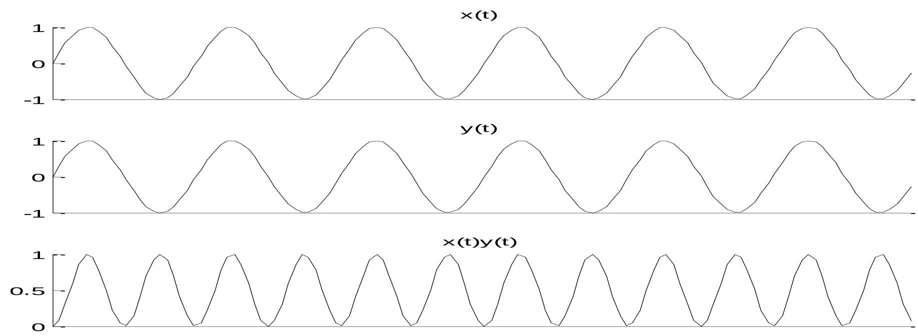
The trend to be fitted will be

$$x_t = a_0 + a_1 t + a_2 t^2 \qquad (7)$$

where t is the time step.

4. Correlation.

In the last section we dealt with trend and its removal. The identification of the trend line was a simple example of correlation – we were estimating that component of the signal which correlated with the passage of time. In a general sense the general idea of correlation is that we are asking the question as to whether two signals are the same, or partially the same. Here *the same* means that they have the same shape within the same time interval. There may be situations where only part of a signal has the same shape as another signal – the signals being *partially correlated*. The test for correlation is essentially a multiplication and integration. If two identical sinusoids are multiplied together then the product waveform will be all positive and the area it includes on integration will be large – the signals are highly correlated. Conversely if the two sinusoids are very different in frequency then the area that results from multiplication, and included in the integration, will be close to zero, taking into account 'positive and negative' parts of the product. This zero integral indicates that the signals are uncorrelated, figure 2. This idea is expressed mathematically in the correlation integral between two time domain signals $x(t)$ and $y(t)$



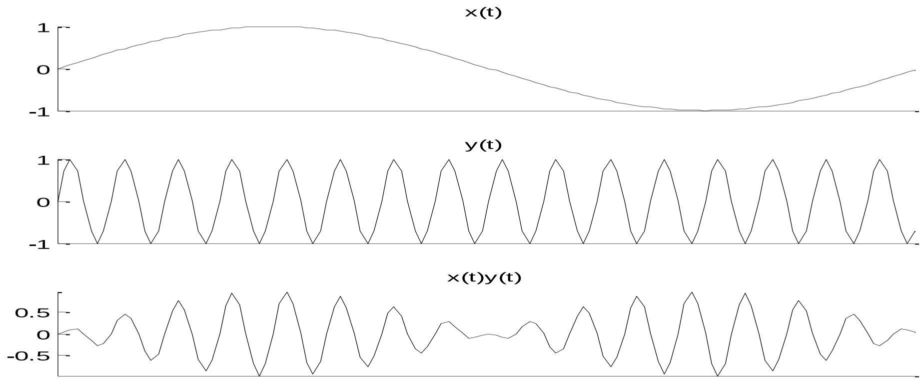


Figure 2. Sinusoids multiplied together – top: waveforms of the same frequency, and bottom: waveforms of different frequencies.

$$r_{xy} = \int_{-T}^{T} x(t)y(t)dt \tag{8a}$$

In practice we normalise r_{xy} so that it has a maximum value of $\pm$ unity. We also remove the mean values of the two signals so that we are only estimating the similarities between non-zero frequency components. For a discrete signal we have

$$r_{xy} = \frac{\sum_{1}^{N}\left[x(n)-\bar{x}\right]\left[y(n)-\bar{y}\right]}{\left[\sum_{1}^{N}\left[x(n)-\bar{x}\right]^2 \sum_{1}^{N}\left[y(n)-\bar{y}\right]^2\right]^{1/2}} \tag{8b}$$

The thus normalised r_{xy} is the *correlation coefficient*. Note that the two denominator terms represent the standard deviations of the two signals except that the division by N has been omitted since it would appear in both numerator and denominator.

4.1 The Cross Correlation Function (CCF).

It is often necessary to establish the correlation between two signals that do not exist over the same time interval. A common example in NDT is a pair of ultrasonic wave signals that have been recorded at different positions on a test specimen and therefore arrive at the recording system at different times. What is done in this case is to calculate the correlation coefficient between the two signals first as they are, then at several different time shifts of one signal with respect to the other. It is a straightforward matter to shift a digitised signal with respect to time. The time shift is known as *lag-time* and the correlation coefficient, now called the *cross correlation function* (CCF), is a function of lag time. Basically, if the two signals are delayed by T seconds with respect to each other but are otherwise identical then the CCF will rise to a maximum value of 1 at lag time = T. Considering first continuous time signals and the unnormalised form of the correlation function we get

$$r_{xy}(\tau) = \int_{T_1}^{T_2} x(t)y(t+\tau)dt \tag{9}$$

Where τ is the lag time. For discrete signals $x(n)$ and $y(n)$, with normalisation to limit the CCF to ± 1 we get

$$r_{xy}(k) = \frac{\sum_{n=1}^{N}\left[x(n)-\bar{x}\right]\left[y(n+k)-\bar{y}\right]}{\left[\sum_{n=1}^{N}\left[x(n)-\bar{x}\right]^2 \sum_{n=1}^{N}\left[y(n)-\bar{y}\right]^2\right]^{\frac{1}{2}}} \qquad (10)$$

Where k is the lag time expressed in sample periods, figure 3.

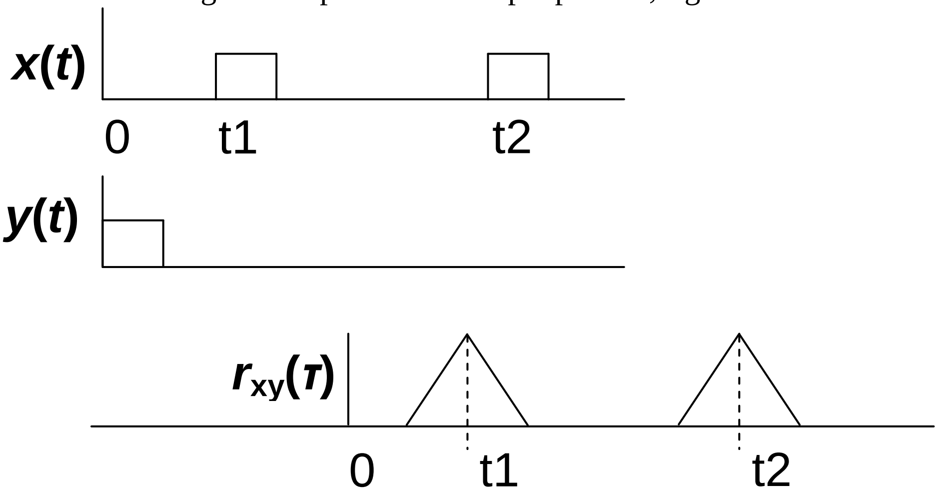


Figure 3. Two simple signals $x(t)$ and $y(t)$, and their cross-correlation function $r_{xy}(\tau)$.

4.2 The Auto Correlation Function (ACF).

The ACF is exactly the same as the CCF except that its two input signals are identical

$$r_{xx}(k) = \frac{\sum_{n=1}^{N}\left[x(n)-\bar{x}\right]\left[x(n+k)-\bar{x}\right]}{\sum_{n=1}^{N}\left[x(n)-\bar{x}\right]^2} \qquad (11)$$

An example is shown on figure 4. The function provides a means to investigate how parts of a signal are the same as other parts, but delayed by k; it identifies periodic trends in a signal. It provides a very basic test for randomness – if the signal is totally uncorrelated with itself at any time lag then it is random and $r_{xy}(k)$ will be unity for k=0 and zero for $k \neq 0$. Although this is true in principle, practical random signals of limited duration are not truly random; a sequence of computer generated random numbers will have an ACF equal to unity at zero lag, but will also show small finite correlations at non-zero lags. These small correlations are known as *self noise*.

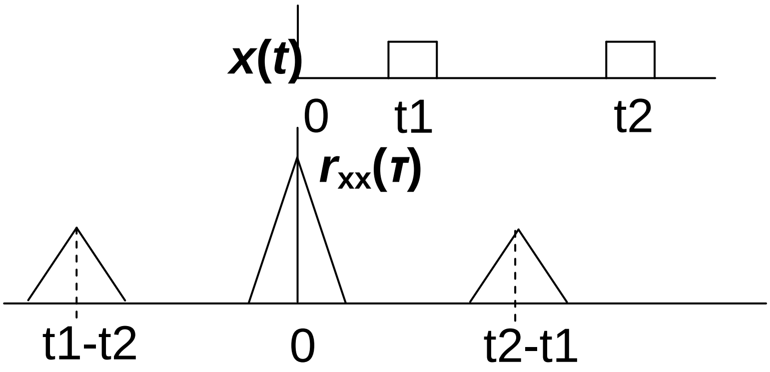


Figure 4. Signal $x(t)$ and its autocorrelation function $r_{xx}(\tau)$.

5. The Frequency Domain.

So far we have considered signals as functions of time; that is, we have analysed signals in the *time domain*. However most signal descriptions and operations, and indeed the processes that generate both, are expressed in the frequency domain. A signal is expressed in terms of its frequency, or if it is a complex signal, in terms of the frequencies it contains. For example a sample of a musical symphony will contain very many more frequencies than the sound from an individual flute. Right across engineering, and certainly in NDT, we regularly (daily, hourly!) require to quantify the frequency content of a complex signal. The notion here is that we can break down a signal into its constituent frequencies and we can assign an amplitude to each of those frequencies. A graph of signal amplitude versus frequency is known as the *signal spectrum*. This idea of breaking a signal down into its constituent components is know as *Fourier decomposition*. We consider first *periodic signals* in continuous time - we wish to express our signal $x(t)$ as the summation of sinewaves of various frequencies. When we do this we arrive at the *Fourier series* (FS) where

$$x(t) = a_0 + \sum_{k=1}^{N} a_k \cos k\omega_0 t + \sum_{k=1}^{N} b_k \sin k\omega_0 t \qquad (12)$$

Points to note about the above expression are:

(i) $\omega_0 = \dfrac{2\pi}{T}$ where T is the period of the original signal; k is the harmonic number which multiplies ω_0 to give the individual component at angular frequency $k\omega_0$. This implies that we only consider that harmonics of the original signal period need to be included in the Fourier Series. It is hoped that the reason for this is obvious – a periodic signal can only be represented by signal components that are themselves periodic with the same periods as that associated with the original signal.

(ii) The signal $x(t)$ is represented by a series of sinewaves as well as a series of cosine waves. The reason for this is that a component of a given frequency in the original signal may not have its zero crossing or maximum value in line with the origin of time used in the original signal.

(iii) The first term a_0 represents the dc or zero frequency component.

Given that we have $x(t)$ or its sampled equivalent in recorded form we now consider how we can calculate the amplitude coefficients a_k and b_k. For this we return to the idea of correlation. We seek similarity of shape between $x(t)$ and our cosines and sines – the question *how much of a given frequency is contained in our signal* $x(t)$ could well be posed as *how similar is the shape of our sine or cosine to* $x(t)$. We seek the correlation coefficient between our sine or cosine and $x(t)$. The procedure to estimate a_k and b_k is therefore one of multiplication of two functions and integrating their product with respect to time. It can be formally demonstrated that

$$a_k = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\cos k\omega_0 t\,dt$$

$$b_k = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\sin k\omega_0 t\,dt$$

$$a_0 = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\,dt \qquad (13)$$

a_0 is the mean value of $x(t)$ over its period T.

It will be realised that cosines and sines are really the same thing, except that their relationship to the origin of time is different – the cosine is symmetric about $t = 0$ (even symmetry) whilst the sine is antisymmetric about $t = 0$ (odd symmetry). The reason for using both is that a real single frequency component in a real engineering signal could have any relationship with respect to $t = 0$. It will be neither all sine or all cosine, although it can be constructed by a combination of the two. We represent this construction on a phase diagram: imagine that we have a component at frequency $k\omega_0$ in which $a_k = b_k = 1$; figure 5 shows the sine and cosine components and their sum. We see that the resultant has sinewave shape, but that it is delayed in time (phase shift). The vector diagram at the bottom of the figure shows this summation where the resultant amplitude c_k is given by

$$c_k = \sqrt{a_k^2 + b_k^2} \qquad (14a)$$

and the phase shift is given by

$$\tan\phi_k = \frac{b_k}{a_k} \qquad (14b)$$

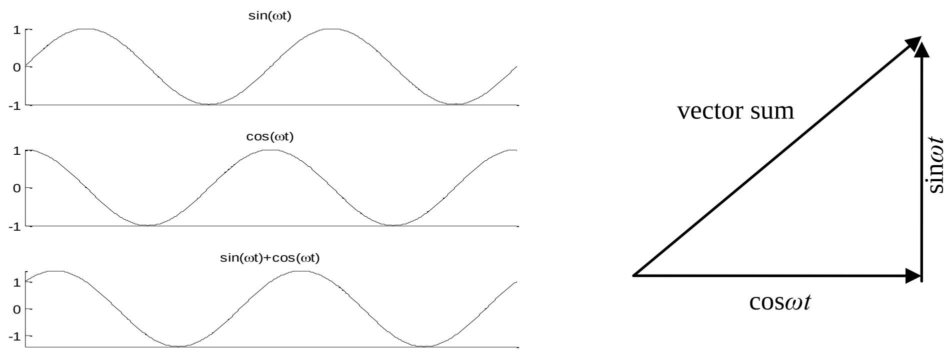


Figure 5. Summation of sine and cosine components and the vector diagram illustrating the operation.

5.1 The Exponential Fourier Series.

The exponential Fourier series (EFS) is essentially the same thing as the trigonometric Fourier series described above. It is more convenient to use mathematically because it uses complex exponentials rather than trigonometric basis functions. When expressed in discrete form it provides for much faster computation than would be possible using sine and cosine functions.

The complex exponential is the most basic signal element used in a vast number of theories and techniques right across engineering, the physical sciences and mathematics. It is expressed as $e^{j\omega t}$ where $j=\sqrt{-1}$, and can be represented on a phasor diagram, figure 6. The idea is that the line representing $e^{j\omega t}$ makes an angle ωt radians with respect to the real axis of the diagram. As time progresses it rotates in an anticlockwise direction. It is known as a *phasor*. The negative phasor $e^{-j\omega t}$ is also shown on the diagram. We can get the relationship between these phasors and the trigonometric functions by simple vector addition, figure 7.

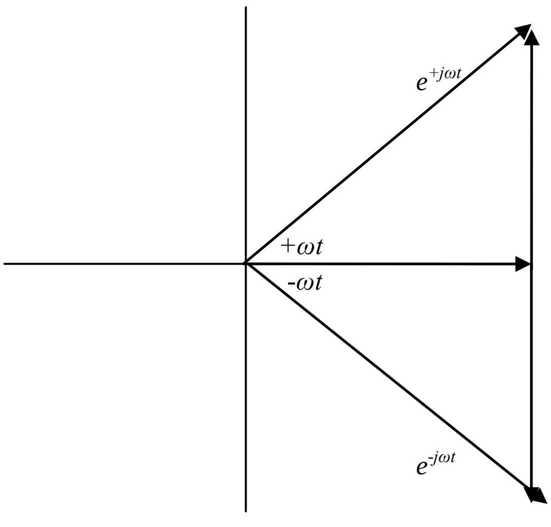


Figure 6. Basic phasor diagram showing the phasor $e^{j\omega t}$.

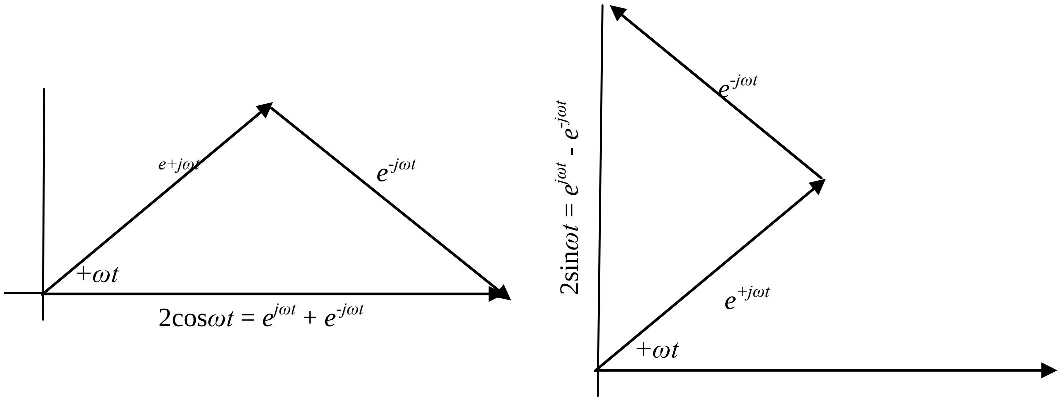


Figure 7. Relationship between basic phasors and the trigonometric functions.

$$e^{j\omega t} + e^{-j\omega t} = 2\cos\omega t$$

$$e^{j\omega t} - e^{-j\omega t} = j2\sin\omega t$$

$$e^{j\omega t} = \cos\omega t + j\sin\omega t$$

$$e^{-j\omega t} = \cos\omega t - j\sin\omega t \qquad (15)$$

The EFS is

$$X(k) = \frac{1}{T}\int_0^T x(t)e^{-jk\omega_0 t}\,dt\,, \qquad (16)$$

which expands to give

$$X(k) = \frac{1}{T}\int_0^T x(t)\cos k\omega_0 t\,dt - j\frac{1}{T}\int_0^T x(t)\sin k\omega_0 t\,dt\,, \qquad (17)$$

or

$$X(k) = X_R(k) + jX_I(k) \qquad (18)$$

Where $X_R(k)$ and $X_I(k)$ are the *real* and *imaginary* parts of $X(k)$. Comparing these with the trigonometric Fourier series coefficients a_k and b_k we see that

$$\begin{aligned} a_k &= 2X_R(k) \\ b_k &= -2X_I(k) \\ c_k &= 2\left|X(k)\right| \end{aligned} \qquad (19)$$

$$\tan\phi_k = \frac{-X_I(k)}{X_R(k)} \qquad (20)$$

These relationships are shown on the argand diagram of figure 8.

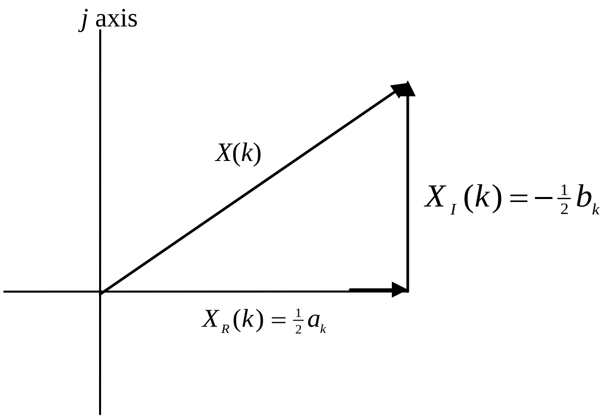


Figure 8. Relationships between the trigonometric and exponential Fourier series coefficients.

The question might be asked as to why the complex exponential in equation 16 is negative. The full explanation is beyond the scope of this course but it stems from correlation theory applied to complex numbers – when the two functions which are to be correlated with each other are multiplied together it is necessary to take the complex conjugate of one of them.

5.2 The Discrete Fourier Series (DFS).

So far, most of our discussion has been in terms of continuous time signals whereas our practice is to process discrete time (digitised) signals. We need to express the ideas of the Fourier Series in terms of discrete mathematics. To give an idea of how this is done, imagine a data array in a computer programme which represents *one period* of a periodic signal. We need to fill the array such that all N elements represent one complete cycle of a cosine – for array points n=0 to N-1 the operation will be

$$x(n) = A \cos n\frac{2\pi}{N} \qquad (21)$$

Where n is the array bin number representing the time step and N is the total number of samples in the array. Similarly, if you wished the array to be filled with k complete cycles, that is, the *kth* harmonic, you would write

$$x(n) = A_k \cos nk \frac{2\pi}{N} \tag{22}$$

Using this idea we can express the continuous time exponential Fourier series of equation 16 in discrete form; we get

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-jnT_s k \frac{\omega_s}{N}} \tag{23}$$

Here, nT_S is the time step, T_S being the time interval between samples. ω_s is the sampling frequency, and ω_s / N represents that frequency that will place one cycle in our array N points long [0 to (N - 1)].

Now

$$\omega_s T_s = 2\pi f_s T_s = 2\pi ,$$

so we have

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-jnk \frac{2\pi}{N}} \tag{24}$$

Many text books use the shorthand notation in which $e^{j\frac{2\pi}{N}}$ is replaced by the term W_N, or sometimes W_N^{-1} . Using the former we get

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) W_N^{-nk} \tag{25}$$

This is the discrete Fourier Series (DFS). Note that $X(k)$ is a complex number where the real part represents half the amplitude of the cosine series and the imaginary part represents (-1) times half the amplitude of the sine series (equations 19). The modulus value of each frequency component, that is its amplitude, is given by

$$|X(k)| = \sqrt{X_R^2(k) + X_I^2(k)} \tag{26}$$

Its phase, representing the balance of cosines and sines is

$$phase\{X(k)\} = \tan^{-1} \frac{X_I(k)}{X_R(k)} \tag{27}$$

We note a possible problem here in that the $\tan^{-1}$ function is limited to $\pm\pi$.

It is very important to note that this series does not include any identification of what the actual time step is, that is, what n corresponds to, nor what actual frequency in Hz is represented by the harmonic variable k. In fact, k merely stands for the number of cycles that would fit into N points in an array.

It should be clear by now that the function of the series is to break down the signal represented by the time series $x(n)$ into its constituent harmonics – the Fourier decomposition principle. The operation inputs a signal in the *time domain* and outputs that signal in the *frequency domain*. In practical situations we frequently require the corresponding operation in the opposite direction – we need to input the frequency domain components of a signal and to output that signal in its time domain form. This is achieved by the inverse discrete Fourier series (IDFS). Each component $X(k)$ represents the complex amplitude of the elemental signal $e^{jnk\frac{2\pi}{N}}$, so to get the time domain we simply add all of these up together

$$x(n) = \sum_{k=0}^{N-1} X(k) e^{jnk\frac{2\pi}{N}} \text{ , or}$$

$$x(n) = \sum_{k=0}^{N-1} X(k) W_N^{nk}$$

where $W_N = e^{j\frac{2\pi}{N}}$. (28)

There is a group of very important computational algorithms which can compute the DFS very rapidly. These are collectively known as the **Fast Fourier Transform (FFT)** and will be discussed shortly, after a brief but important consideration of the structure of the DFS.

5.3 Symmetries in the DFS.

We can imagine $X(k)$ set out in two graphs – one for its imaginary part and one for its real part, figure 9. The component at $k = 0$ is merely

$$X(0) = \frac{1}{N} \sum_{n=0}^{N-1} x(n), \tag{29}$$

that is, the mean or dc value of the signal.

Now imagine components outside of the range $k = 0$ to $k = N$-1 spaced N apart. We set $k = k \pm pN$ where p is an integer and obtain

$$X(k \pm pN) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j(k \pm pN)n\frac{2\pi}{N}}$$

$$X(k \pm pN) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-jkn\frac{2\pi}{N}} e^{\mp jpNn\frac{2\pi}{N}} . \tag{30}$$

The last term is always unity so we have

$$X(k \pm pN) = X(k) \tag{31}$$

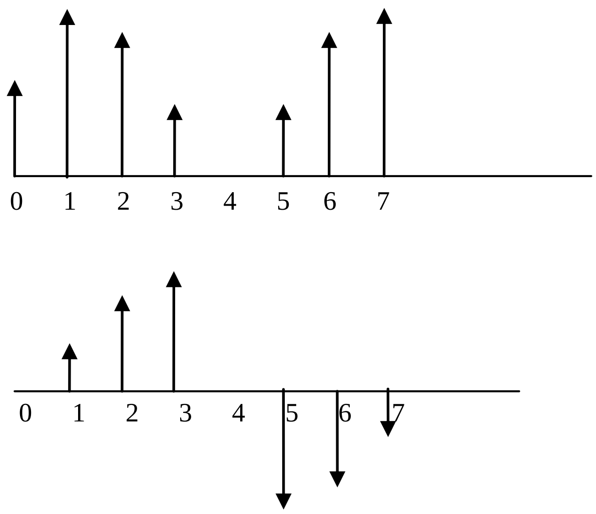


Figure 9. Real (top) and imaginary (bottom) parts of $X(k)$.

That is to say, the series repeats identically at intervals of N – It is periodic in the frequency domain with period N. We note here that the Nth frequency component corresponds to N cycles in our array N points long; ***N* thus corresponds to the sampling frequency** and the series can be said to repeat identically at intervals of the sampling frequency, figure 10.

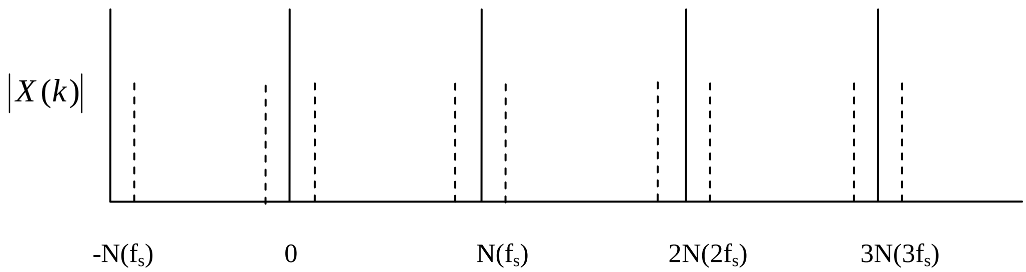


Figure 10. Symmetry in the DFS: The function repeats at intervals of the sampling frequency.

We now consider symmetry within each period, on either side of $k = 0, \pm N, \pm 2N$ etc.

$$X(\pm pN - k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n) e^{-j(\pm pN - k)n\frac{2\pi}{N}}$$

$$X(\pm pN - k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n) e^{+jnk\frac{2\pi}{N}} e^{\mp jpNn\frac{2\pi}{N}} \qquad (32)$$

As in the last example, the final term is unity whereas the 'active' complex exponential term is $e^{+jnk\frac{2\pi}{N}}$. The sign of the exponent has changed from *minus* to *plus* and the implication of this is complex symmetry within the interval 0 to *N*, figure 11.

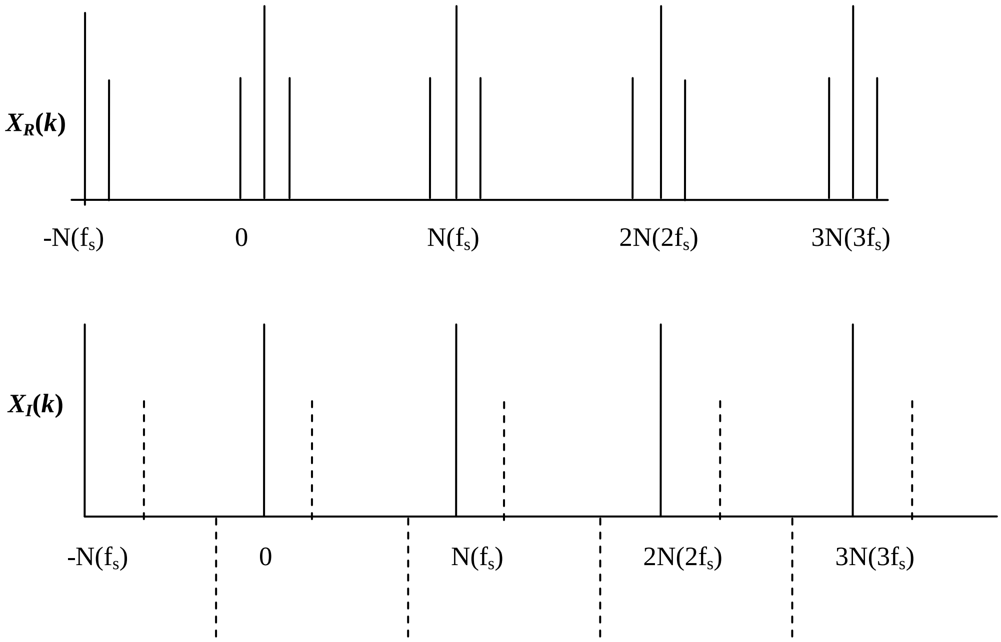


Figure 11. Complex conjugate symmetry within the DFS.

We note that $X(0)$ is wholly real and its imaginary part is zero; this will be true for $X(N)$, $X(2N)$, $X(3N)$ and so on. This complex conjugate symmetry is related to the sampling theorem. It implies that frequency components in the range 0 to $N/2$ (and pN to $(pN + N/2)$) repeat as their complex conjugates in the range N to $(N - N/2)$ and pN to $pN – N/2$. The DFS cannot therefore determine frequency components above $N/2$, equivalent to half of the data sampling frequency. Viewed the other way around, a signal containing components at a maximum frequency f_{MAX} must be sampled at a frequency of at least $2f_{MAX}$. The *aliassing problem* referred to earlier in these notes can be interpreted as the two halves of this complex conjugate symmetry overlapping each other somewhere in the centre of the data array, figure 12.

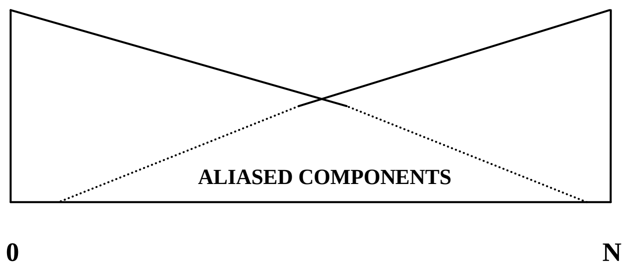


Figure 12. Frequency spectrum of an aliased signal.

Finally, we note that $X(k)$ really only has N unique numbers, up to $N/2$ for both the real and imaginary components; it has N degrees of freedom. The original time series $x(n)$ also only had N unique numbers – so information is neither gained nor lost in the transformations from *time to frequency* or *frequency to time.*

5.4 The Fast Fourier Transform (FFT).

The term FFT is a collective one which covers a large number of computational algorithms designed for the efficient calculation of the DFS and its inverse. Detailed descriptions of how FFTs work is beyond the scope of this course so we will describe what they do, and how they do it, from the user's perspective.

First, the forward FFT requires as its input an array of time domain data N points long. This array is treated as though it was a single period of a periodic signal, although in practice experimental data is not generally periodic over all time in this sense. The FFT also needs a second array of the same length (N points). ***N* must be an integer power of 2**. The reason for this is that the FFT breaks down the computation of $X(k)$ into several 'sub' FFTs of length $N/2$, $N/4$, $N/8$ and so on, the limit being the so-called butterfly operation which is an $N = 2$ calculation. Not all FFTs break down the computation as far as $N = 2$. They may stop at, for example, $N = 32, 16, 8, 4$ or 2. The point at which they stop is known as the *radix* of the transform – a *radix 4* operation breaking down as far as $N = 4$. The process of breaking down is known as *decimation.*

Actual software calls to FFT routines differ one to another, but a basic operation will include, somewhere, the following.

- Set up two arrays

$A(n)$	0 to N-1	or	1 to N
$B(n)$	0 to N-1	or	1 to N

- Copy the time domain signal to be analysed into array $A(n)$, sometimes known as the *real array*.

- Set all of the elements of array $B(n)$ to zero; this array is sometimes known as the *imaginary array*.

- 'Tell' the FFT how many points it is working on; that is, set the value of N.

- 'Tell' the FFT which direction it is going in; this is usually achieved either by setting an integer to +1 to represent time to frequency (forward) transformation or to –1 to represent frequency to time (inverse or backward) transformation. Sometimes this is achieved by setting the sign of N ; it may also be achieved using routines with different names for the forward and inverse operations, such as FFT and IFFT.

When the FFT runs in the forward direction the arrays $A(n)$ and $B(n)$ are overwritten and end up containing the real and imaginary parts of $X(k)$ between k=0 and k=N-1.

After the run $A(0)$ will contain the dc term (average over the period), $A(1)$ the first harmonic cosine amplitude, $A(2)$ the second harmonic cosine amplitude and so on up to $k = \frac{N}{2} - 1$. $A\left(\frac{N}{2}\right)$ should be zero (half the sampling frequency). The rest of the array up to $A(N$-1) will be the mirror image of the first half, following the expected symmetry discussed earlier.

In the imaginary array $B(0)$ should be zero and the right hand half should be the negative of the mirror image of the left hand half. Figure 13 illustrates this operation for an 8 point sequence.

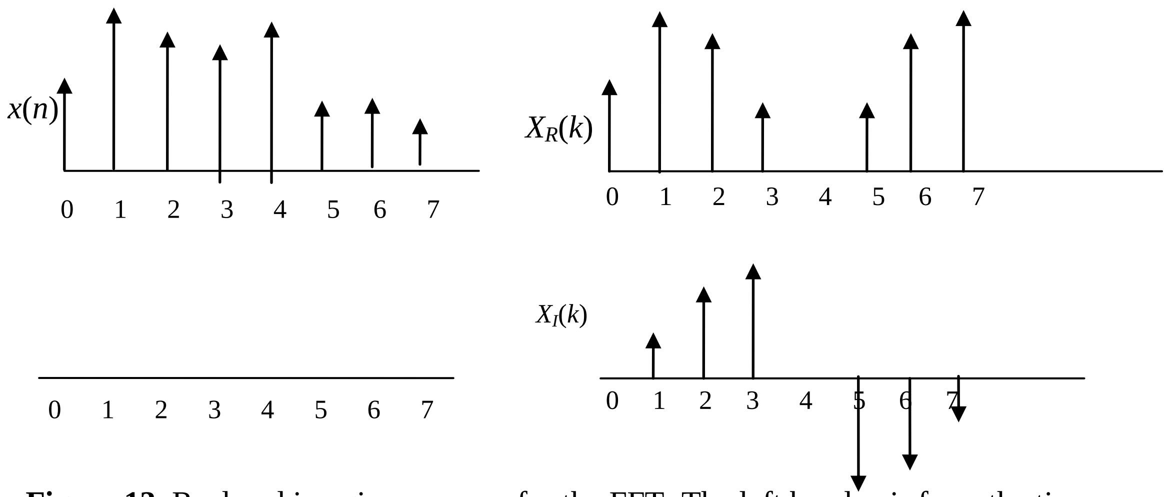


Figure 13. Real and imaginary arrays for the FFT: The left hand pair form the time domain input and the right hand pair form the frequency domain output. The top pair are the real array, and the bottom pair are the imaginary array.

The *inverse FFT* is a little more complicated. We have to fill arrays A and B with the real and imaginary parts of our frequency domain signal taking account of the required complex conjugate symmetry between the left hand and right hand parts of the array. The following operations are required:

If the array indices run from k=0 to k=N-1

Set $A(0)$ = dc value
$B(0) = 0$
$A\left(\frac{N}{2}\right) = 0$
$B\left(\frac{N}{2}\right) = 0$

Fill $A(1)$ to $A\left(\frac{N}{2} - 1\right)$
$B(1)$ to $B\left(\frac{N}{2} - 1\right)$

For k=1 to $k = \frac{N}{2} - 1$

Set $A(N - k) = A(k)$

$$B(N-k) = -B(k)$$

Then run the inverse transform.

If the array indices run from *k*=1 to *k*=*N*

Set $A(1)$ = dc value
$B(1)$ = 0
$A\left(\frac{N}{2}+1\right)=0$
$B\left(\frac{N}{2}+1\right)=0$

Fill $A(2)$ to $A\left(\frac{N}{2}\right)$
$B(2)$ to $B\left(\frac{N}{2}\right)$

For k=2 to $k=\frac{N}{2}$

Set $A(N-k+2)=A(k)$
$B(N-k+2)=-B(k)$

When which ever is appropriate of these two operations is completed, run the FFT in inverse mode. After the run the real array $A(n)$ will contain the required time domain record and the imaginary array $B(n)$ should be zero throughout. Note here that we have used the convention that the frequency index is k, whilst the time step index is n.

Finally, the computational speed of the FFT is very much greater than the speed of the DFS, and a very basic calculation shows this. An N - point DFS will require N^2 complex multiply and add operations whereas a radix –2 FFT will require $N \log_2 N$ operations. The table below shows the number of operations for the DFS and FFT as N is increased

N	N^2	$N\log_2 N$
2	4	2
4	16	8
8	64	24
16	256	64
32	1024	160
64	4096	384
128	16384	896

The speed advantage of the FFT should be obvious.

5.5 Using the FFT - 1: Spectral Analysis.

Spectral analysis merely means calculating the amplitude of the frequency components in a signal. A time series $x(n)$ is recorded and its FFT computed. The amplitude spectrum of the signal is simply the modulus of the FFT output. The power spectrum is the square of the modulus, so for the amplitude spectrum we have

$$|X(k)| = \sqrt{X_R^2(k) + X_I^2(k)} \tag{33a}$$

and for the power spectrum we have

$$|X(k)|^2 = X_R^2(k) + X_I^2(k) \tag{33b}$$

When plotting these values as a graph it makes sense only to plot up to $k = \frac{N}{2}$ to avoid the symmetry components. If the original data was sampled at frequency f_s then the x-axis of the graph should be marked in frequency units up to $f_s/2$.

There may be circumstances where the calculated spectrum needs to be calibrated in some way, and here we use as an example the measurement of the ultrasonic absorption coefficient in a test material. An ultrasonic wave recorded in such an experiment could take the form

$$|X(z, f)| = |S(f)| e^{-\alpha(f) z} \tag{34}$$

$X(z,f)$ is the frequency domain signal after travelling distance z in the test material and $\alpha(f)$ is the attenuation coefficient which we wish to find; f is frequency. $S(f)$ is the unknown response of the electronic system, the transducers, and other aspects of the apparatus. To find $S(f)$ we do another experiment in which $\alpha(f)$ is close to zero in order to get a calibration signal equal to $S(f)$.

The operation we perform is to record two signals in the time domain: $x(t)$ corresponding to the test system and the test material, and $s(t)$ corresponding to the system response. We take the modulus FFTs of these two signals to get $X(z,f)$ and $S(f)$. We divide these one by the other, frequency by frequency, to get

$$e^{-\alpha(f) z} = \frac{|X(z, f)|}{|S(f)|} = Y(f) \tag{35}$$

This operation can be limited to $\frac{N}{2}$ points. We need to protect the division against low denominator values, and a good way to achieve this is to set the result to zero if any frequency component falls below a threshold value, typically 0.001 times the maximum value in the array. The attenuation coefficient is then calculated thus

$$\alpha(f) = \frac{1}{z} \ln \frac{|S(f)|}{|X(z, f)|} \tag{36}$$

5.6 Using the FFT - 2: Frequency Domain Filters.

A convenient way to filter a signal is to calculate its FFT and then to multiply both $X_R(k)$ and $X_I(k)$ by a window function which reduces the amplitude of unwanted frequency components to zero or near zero. In principle this could be achieved by applying a window such as $W_1(k)$ shown on figure 14. However, we shall see later that this results in a distorted output – due to the sharp transitions in its shape. Instead we use window functions that have gently rising or falling tapers at their transitions, $W_2(k)$. When the window has been applied to the real and imaginary parts of $X(k)$ up to $k = \frac{N}{2}$ we rebuild the symmetry components in the right hand side of the array and then take the inverse FFT to get the filtered result in the time domain.

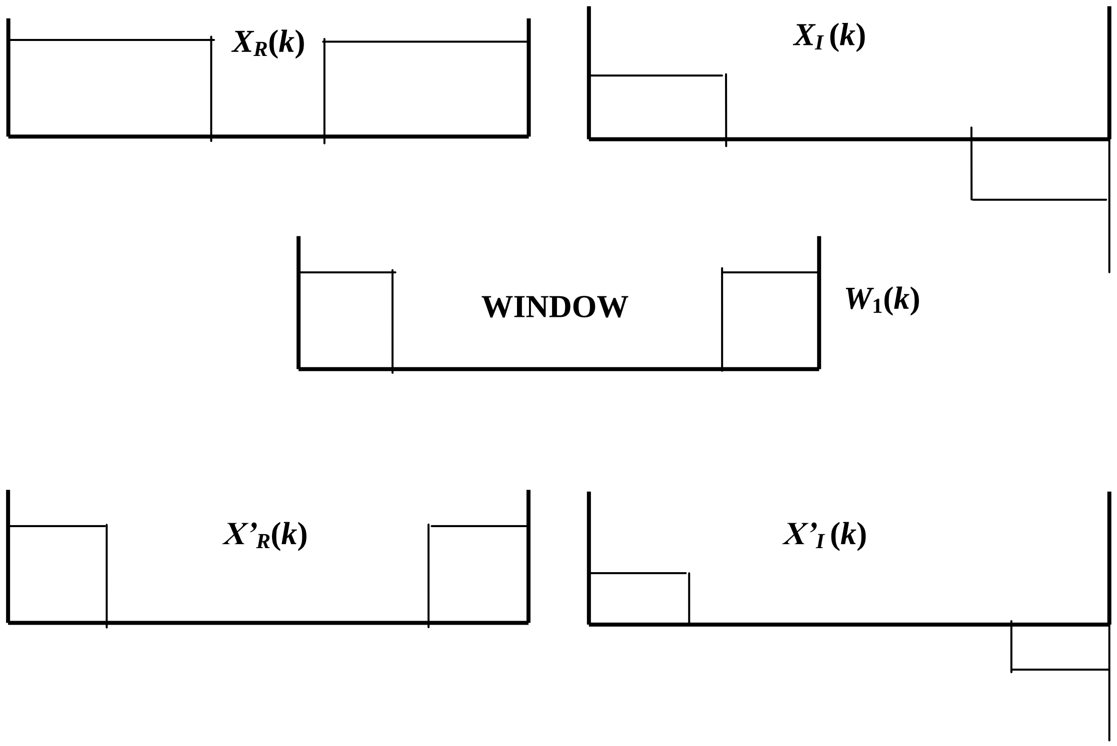


Figure 14. Multiplication of FFT by window functions.

The question arises as to what mathematical shape the window should be – it is often stated that there are as many types of window as there are joiners in the world! A safe option is to use the so-called *hanning window* (after its proposer Julius von Hann) which takes the form of a decaying cosine. The window that is to be multiplied by our frequency domain transform of our input data is:

Low frequency stop band between $k = 0$ **to** $k = k_1 - 1$.

$W(k) = 0$

Rising transition between $k = k_1$ **to** $k = k_2$

$$W(k) = \frac{1}{2}\left[1 - \cos\left[\frac{(k - k_1)}{(k_2 - k_1)}\right]\pi\right]$$

Pass band between $k = k_2 + 1$ **to** $k = k_3 - 1$

$W(k) = 1$

Falling transition between $k = k_3$ **to** $k = k_4$

$$W(k)=\frac{1}{2}\left[1+\cos\left[\frac{(k-k_3)}{(k_4-k_3)}\right]\pi\right]$$

High frequency stop band between $k = k_4 + 1$ to $k = \frac{N}{2}$

$$W(k) = 0 \qquad (37)$$

Once the window is applied to all signal components up to $\frac{N}{2}$ the right hand half of the array is built up to reflect the required complex conjugate symmetry.

5.7. Using the FFT – 3: Digital Versions of Electric Analogue Filters.

A simple lowpass resistance-capacitance (RC) filter is shown on figure 15. The input signal in the time domain is $x(t)$ with a spectrum $X(\omega)$; the output is $y(t)$ with a spectrum $Y(\omega)$. $Y(\omega)$ and $X(\omega)$ are related by the frequency response of the filter, $H(\omega)$.

$$Y(\omega)=H(\omega)X(\omega) \qquad (38)$$

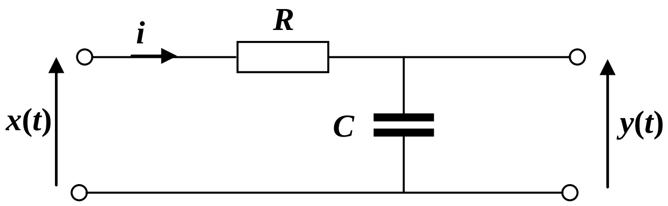


Figure 15. First order lowpass electric filter.

In order to mimic this operation digitally we need to record $x(t)$ and calculate its FFT, $X(k)$ equivalent to $X(\omega)$ above. We then express $H(\omega)$ as a frequency domain window $H(k)$ and multiply it by $X(k)$ up to $k=\frac{N}{2}$ to get $Y(k)$. We then build the complex conjugate of $Y(k)$ in the right hand side of the frequency domain data arrays (real and imaginary). The final step is to take the inverse FFT to get the output of the filter in the time domain, $y(t)$. The difficult part is getting the digital version of the frequency response $H(k)$ from our knowledge of the 'analogue' response $H(\omega)$. The frequency response of our simple analogue filter is given by

$$H(\omega)=\frac{\omega_0}{\omega_0+j\omega} \qquad (39a).$$

Here ω_0 is the cut off frequency of the filter. Now if the data sampling frequency is ω_s and we have N samples in our FFT the value of k which corresponds to ω_0 is

$$k_0=\frac{\omega_0}{\omega_s}N \qquad (39b)$$

All other frequencies are represented by

$$k = \frac{\omega}{\omega_s} N \qquad (39c)$$

$H(\omega)$ can be rearranged to give

$$H(\omega) = \frac{\dfrac{\omega_0}{\omega_s}}{\dfrac{\omega_0}{\omega_s} + j\dfrac{\omega}{\omega_s}} \qquad (39d)$$

Whence

$$H(k) = \frac{k_0}{k_0 + jk} \qquad (40)$$

The real and imaginary parts of $H(k)$ can be found by multiplying numerator and denominator by $(k_0 - jk)$, whence

$$H(k) = \frac{k_0(k_0 - jk)}{k_0^2 + k^2} = \frac{1 - j\dfrac{k}{k_0}}{1 + \dfrac{k^2}{k_0^2}} \qquad (41)$$

So, we have the real and imaginary parts

$$H(k) = H_R(k) + jH_I(k) \qquad (42a)$$

$$H_R(k) = \frac{1}{1 + \dfrac{k^2}{k_0^2}} \qquad (42b)$$

$$H_I(k) = \frac{-\dfrac{k}{k_0}}{1 + \dfrac{k^2}{k_0^2}} \qquad (42c)$$

We multiply this by the spectrum of $x(t)$ to get $Y(k)$, up to $k = \frac{N}{2}$.

$$Y(k) = \left[H_R(k) + jH_I(k)\right]\left[X_R(k) + jX_I(k)\right], \qquad (43a)$$

whence

$$Y_R(k) = H_R(k)X_R(k) - H_I(k)X_I(k) \qquad (43b)$$

$$Y_I(k) = H_I(k)X_R(k) + H_R(k)X_I(k) \qquad (43c)$$

$Y(k)$ is then reflected for symmetry in the right hand half of the array, between $k = \frac{N}{2} + 1$ and $k = N - 1$. The inverse FFT then results in the output time series $y(n)$.

Whilst we have used a very basic electric filter in this example the same method can be used for filters of any complexity. The advantage is that the digital filter frequency response only needs to be computed once.

5.8 The Fourier Transform in the Context of Analogue Signals.

Our discussion so far has focussed on signals that were periodic in time, or assumed to be so for the computational convenience of using the FFT, for example. When we deal with analogue electrical systems, such as filters, it is useful to use a mathematical calculus that does not assume that they are periodic in time or frequency. The *Fourier Transform* is the mathematical operation that transforms a time domain signal into the frequency domain, with no assumptions as to periodicity. We touched on this notion when a simple electric filter was introduced in section 5.7. The Fourier transform of a time domain signal is given by

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\, dt \qquad (44)$$

Clearly, this equation is similar to the exponential Fourier series except that the frequency variable ω is continuous rather than an integer multiple of the frequency that corresponds to one cycle in a period. It has an inverse by which $x(t)$ can be obtained from $X(\omega)$ – transformation from the frequency domain back to the time domain

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}\, d\omega \qquad (45)$$

The transform $X(\omega)$ is complex – its real part represents signals of the form $\cos\omega t$ whilst its imaginary part represents signals of the form $\sin\omega t$.

$$X(\omega) = X_R(\omega) + jX_I(\omega) \qquad (46)$$

We note that the transformation is achieved by the correlation process – multiplication and integration, as before. Fourier transforms are important in the manipulation and understanding of the responses of linear systems. They have many important properties, but due to the brevity of this course we will address only the most important ones here.

Linearity implies that if two signals are added together then the Fourier transform of their sum will be equal to the sums of their two individual Fourier transforms.

$$\begin{aligned} x(t) &= x_1(t) + x_2(t) \\ X(\omega) &= X_1(\omega) + X_2(\omega) \end{aligned} \qquad (47)$$

Scaling follows from linearity:

$$x(t) = ax_1(t) + bx_2(t)$$
$$X(\omega) = aX_1(\omega) + bX_2(\omega) \tag{48}$$

Differentiation in the time domain

If $y(t) = \dfrac{dx(t)}{dt}$

We can write $x(t)$ as the inverse of its Fourier transform

$$y(t) = \frac{d}{dt}\left[\frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega\right] \quad = \quad \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)\frac{d}{dt}e^{j\omega t}d\omega$$

It follows that

$$y(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}[\,j\omega X(\omega)]e^{j\omega t}d\omega$$

Thus $y(t) = \dfrac{dx(t)}{dt} \Leftrightarrow j\omega X(\omega)$ (49)

So, differentiation in the time domain implies multiplication by $j\omega$ in the frequency domain; this is a highpass filter process.

Integration in the time domain

$$y(t) = \int x(t)dt \quad = \quad \frac{1}{2\pi}\iint X(\omega)e^{j\omega t}d\omega dt \quad = \quad \frac{1}{2\pi}\int X(\omega)\frac{e^{j\omega t}}{j\omega}d\omega$$

Thus $y(t) \Leftrightarrow \dfrac{X(\omega)}{j\omega}$ (50)

That is, integration in the time domain implies division by $j\omega$ in the frequency domain.

Time shift is important because many of the signals we study are shifted in time with respect to some notional origin of time - such as the left hand side of a graticule on an oscilloscope. The time shift is expressed algebraically as

$$y(t) = x(t-\tau)\,, \tag{51}$$

Where τ is the time shift – because τ is preceded by a minus sign the shift represents a delay to $x(t)$ of τ seconds. We require the Fourier transform of $y(t)$

$$Y(\omega) = \int y(t)e^{-j\omega t}dt \quad = \quad \int x(t-\tau)e^{-j\omega t}dt \tag{52}$$

We use a 'trick' here – we invoke a new variable, $p = t - \tau$, so that $t = p + \tau$, and

$dt = dp$, τ being fixed. We then rewrite the integral

$$Y(\omega) = \int x(p) e^{-j\omega(p+\tau)} dp \quad = \quad \int x(p) e^{-j\omega p} dp \bullet e^{-j\omega\tau} \qquad (53)$$

The integral is simply the Fourier transform of $x(p)$ which equals the Fourier transform of $x(t)$ multiplied by $e^{-j\omega\tau}$.

$$Y(\omega) = e^{-j\omega\tau} X(\omega) \qquad (54)$$

The units of $\omega\tau$ are angular (radians, where π radians = 180º). The multiplier $e^{-j\omega\tau}$ applies a *phase shift* to $X(\omega)$ – it changes the balance of sines and cosines in the transform so that the shifted signal is built up by adding appropriate sines and cosines in such a manner that it ends up in the right place in the time domain – at τ seconds from the origin.

6. A Look at Linear System Responses.

We take the words *linear system* to cover such things as electric filters, paths for electric current, or wave propagation media. These systems will have *responses* by which we mean that when a signal passes through the system the output will differ from the input in that some frequencies in the signal will have been emphasised or de-emphasised by the system. That is to say, they will have been *filtered*. This process is quantified by the *frequency response* of the system. Figure 16 shows a system with a frequency response $H(\omega)$. The input signal is $x(t)$ as a function of time with the Fourier transform $X(\omega)$ and the output signal is $y(t)$ with the Fourier transform $Y(\omega)$. In figure 16 the two-way arrow represents Fourier transformation.

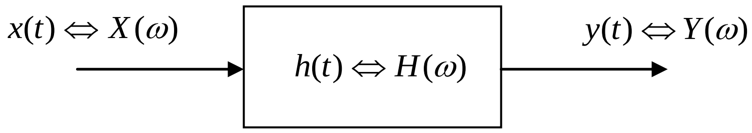


Figure 16. A linear system.

The output response is obtained from the input response thus

$$Y(\omega) = H(\omega) X(\omega) \qquad (55)$$

We have seen this before in the context of using an FFT to simulate the action of a simple electric filter. Beginning in the time domain the filtering pocess can be conceived of as follows

Input signal	$x(t)$
Take Fourier transform	$x(t) \Rightarrow X(\omega)$
Filter in frequency domain	$Y(\omega) = H(\omega) X(\omega)$
Return to the time domain	$Y(\omega) \Rightarrow y(t)$

Throughout engineering and physics it is often useful to consider the filtering operation in the time domain alone – but how do we express the filter response $H(\omega)$ as a time domain function? We achieve this through the dual concept of the *unit impulse* and the system *impulse response*.

6.1 The Unit Impulse and the Impulse Response.

The unit impulse, also known as the *delta function* or the *Dirac function* can be conceived of as a short pulse whose area is unity, figure 17.

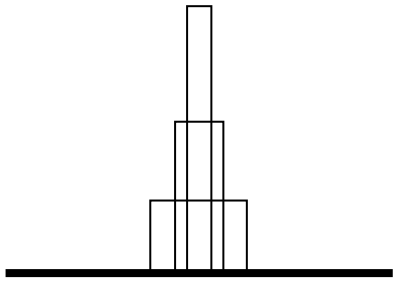


Figure 17. The unit impulse as a limiting case of pulses of unit area.

We can make the pulse thinner and taller, retaining its area at unity – $\Delta T/2$ wide by $2/\Delta T$ high, for example. If we continue this process we end up with a pulse of zero width and infinite height. This is the *unit impulse* $\delta(t)$ with an area or *weight* of unity, expressed by the following integral

$$\int_{-\infty}^{\infty} \delta(t)dt = 1. \qquad (56)$$

The unit impulse is an important signal because it contains all frequencies at the equal amplitude of unity. Its Fourier transform is

$$\Delta(\omega) = \int_{-\infty}^{\infty} \delta(t)e^{-j\omega t}dt. \qquad (57a)$$

Because $\delta(t)$ only exists at $t = 0$ the term $e^{-j\omega t}$ is equivalent to e^0 = unity. Thus

$$\Delta(\omega) = 1. \qquad (57b)$$

If we use $\delta(t)$ as the input to a linear system we are effectively inputting a signal that contains all frequencies at equal amplitude at the same time. We would thus expect the output of the system to represent the frequency response of the system in some way. This representative response is known as the *impulse response* of the system $h(t)$. It is related to the frequency response through the Fourier transform

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{-j\omega t}dt. \qquad (58)$$

The question arises now as to whether $h(t)$ can operate on $x(t)$ directly to get $y(t)$ – indeed, it can, and the operation is known as *convolution*.

6.2 Convolution in the time Domain.

For this discussion we concentrate here on the convolution operation as it applies to time domain signals. Convolution in the frequency domain is also an important operation, and we will discuss this later in the course. For clarity and convenience it is easiest to consider convolution in terms of sampled signals. Figure 18a illustrates a continuous time system and its impulse response, whilst figure 18b shows the sampled version.

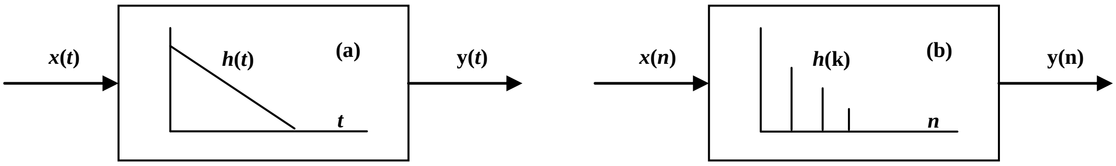


Figure 18. Continuous time (a) and sampled (b) system responses.

Figure 19 shows what happens to the input signal $x(n)$. Each 'impulse' or sample in the signal (x_0 x_1 x_2 etc) will cause the impulse response to appear at the system output, *weighted* by the amplitude of the individual signal sample. The system output will be the *summation* or *superposition* of all of these responses. We can build up $y(n)$ thus

$$
\begin{aligned}
y_0 &= x_0 h_0 \\
y_1 &= x_0 h_1 + x_1 h_0 \\
y_2 &= x_0 h_2 + x_1 h_1 + x_2 h_0 \\
y_3 &= x_0 h_3 + x_1 h_2 + x_2 h_1 + x_3 h_0
\end{aligned}
\tag{59}
$$

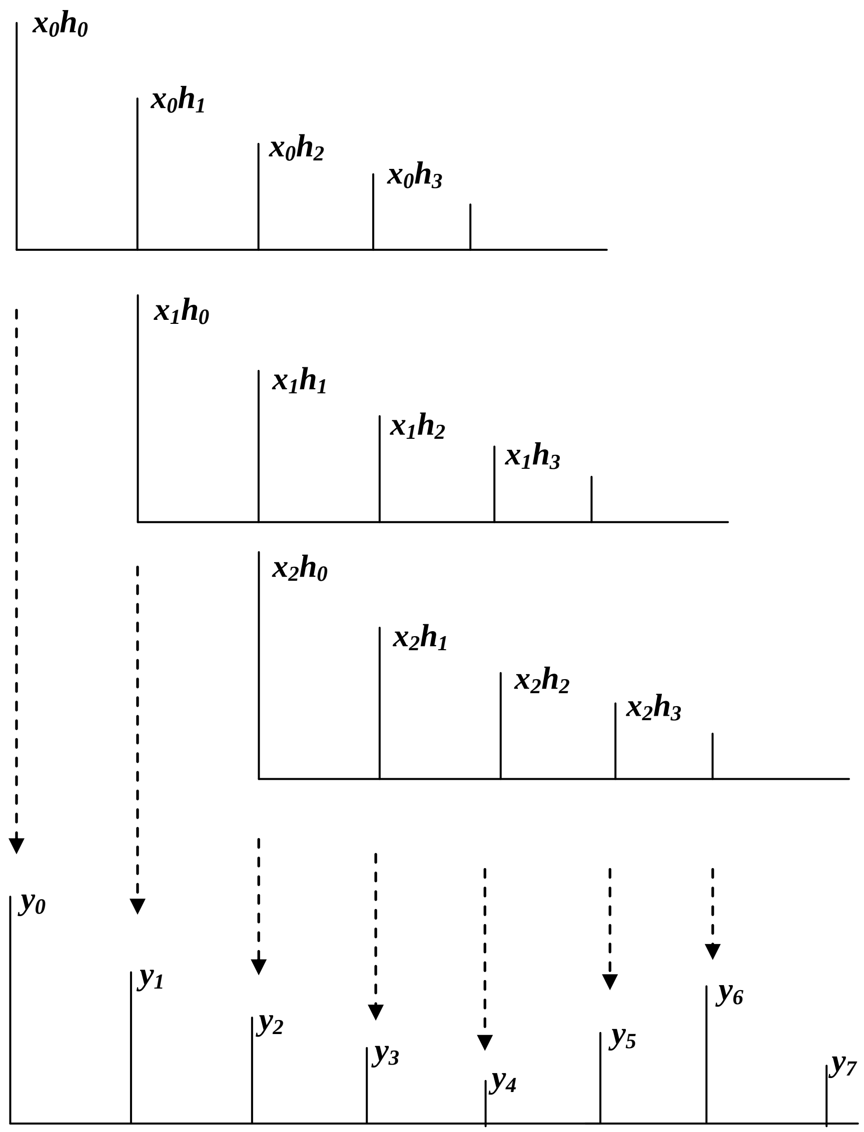


Figure 19. Successive impulse responses at system output.

Note here that for clarity a subscript notation has been use to indicate the time step - $x_n \equiv x(n)$. There is a clear pattern in these responses, and this can be expressed mathematically as

$$y_n = \sum x_{n-k} h_k$$

or, equivalently

$$y(n) = \sum x(n-k)h(k) \qquad (60)$$

These expressions represent what is known as the *convolution sum*. We can recast the way we think about the convolution operation by a graphical representation of this operation. Figure 20 shows a sampled signal x_n; imagine that we wish to calculate the 4^{th} output y_4.

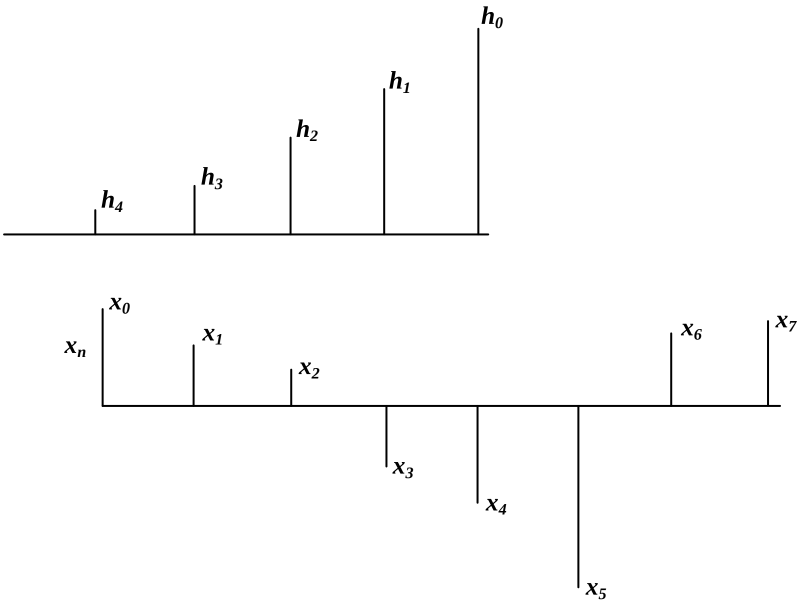


Figure 20. The convolution operation.

We take the impulse response and lay it out backwards, beginning at x_4. The output value y_4 is calculated by multiplying the impulses that are opposite each other

$$y_4 = h_0 x_4 + h_1 x_3 + h_2 x_2 + \ldots\ldots \tag{61a}$$

and so on.

The next output, y_5, is calculated by moving the impulse response one step to the right and repeating the process.

$$y_5 = h_0 x_5 + h_1 x_4 + h_2 x_3 \ldots\ldots\ldots\ldots \tag{61b}$$

The convolution operation can easily be expressed in continuous time by imagining that our samples in both the signals and the impulse responses are brought closer and closer together – we increase the sampling rate until it approaches infinity. The operation becomes

$$y(t) = \int_{-\infty}^{\infty} x(t-\tau) h(\tau) d\tau \tag{62a}$$

This is the *convolution integral*, which is sometimes expressed by the operator *, as follows

$$y(t) = x(t) * h(t)\,. \tag{62b}$$

The continuous time operation is shown graphically on figure 21.

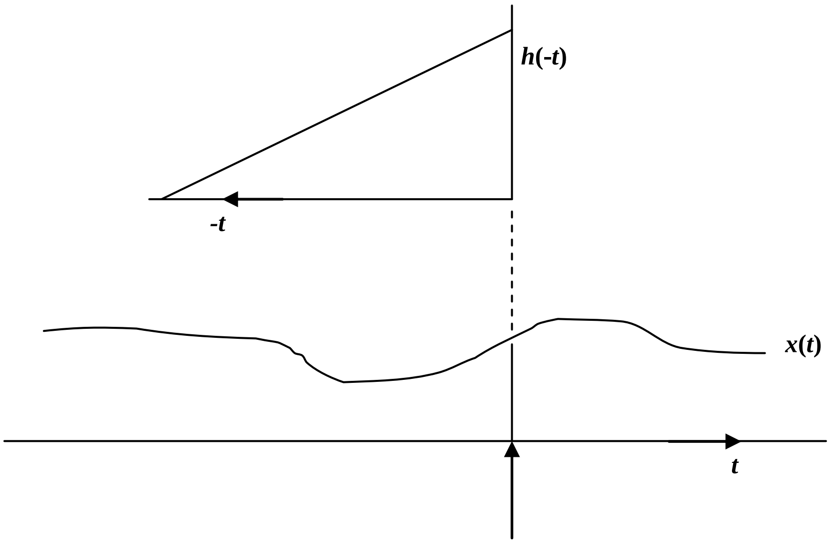


Figure 21. Convolution in continuous time.

7. Frequency Domain – Time Domain Equivalences.

It will be clear from the above that we have expressed our filtering operation in the time domain as a convolution, and in the frequency domain as a multiplication. A similar consideration applies to the operation of correlation that we discussed earlier in this course. In order to 'bring all of this together' this section deals formally with these domain equivalences and shows how they can be put to good use when we use the FFT to carry out filtering operations.

7.1. Time Domain Convolution and the Fourier Transform.

We have already derived the output $y(t)$ of a system from its input $x(t)$ in convolution with the system impulse response $h(t)$.

$$y(t) = \int_{-\infty}^{\infty} x(t-\tau)h(\tau)d\tau$$

To arrive at the frequency domain operation we take the Fourier transform

$$Y(\omega) = \int_{-\infty}^{\infty} y(t)e^{-j\omega t}dt$$

or

$$Y(\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(t-\tau)h(\tau)d\tau\, e^{-j\omega t}dt \qquad (63a)$$

As both of these integrals are linear we can change the order of integration to get

$$Y(\omega) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} x(t-\tau)e^{-j\omega t}dt \right] h(\tau)d\tau \qquad (63b)$$

We invoke time shift theorem to get

$$Y(\omega) = \int_{-\infty}^{\infty} e^{-j\omega\tau} X(\omega) h(\tau) d\tau \quad = \quad X(\omega) \int_{-\infty}^{\infty} h(\tau) e^{-j\omega\tau} d\tau \qquad (63c)$$

That is to say

$$Y(\omega) = X(\omega) H(\omega) \qquad (63d)$$

This proves that convolution in the time domain is equivalent to multiplication in the frequency domain.

7.2 Time Domain Correlation and the Fourier Transform.

Recall that the kernel of the cross correlation operation is the following integral

$$r_{xy}(\tau) = \int_{-\infty}^{\infty} x(t) y(t+\tau) dt \qquad (64)$$

This is very similar to the convolution integral and so cross correlation would thus be expected to have an equivalent operation in the frequency domain, and we derive this here. The Fourier transform of $r_{xy}(\tau)$ is

$$R_{xy}(\omega) = \int r_{xy}(\tau) e^{-j\omega\tau} d\tau \quad = \quad \int \int x(t) y(t+\tau) dt \, e^{-j\omega\tau} d\tau \qquad (65)$$

Consider the term

$$\int y(t+\tau) e^{-j\omega\tau} d\tau \quad = \quad e^{j\omega t} Y(\omega) \qquad (66)$$

This is just the Fourier transform of $y(\tau)$ with the time shift theorem applied. Substituting this back into equation 65 gives

$$R_{xy}(\omega) = Y(\omega) \int x(t) e^{+j\omega t} dt \qquad (67)$$

Now the integral term in the above equation is the Fourier transform of $x(t)$ except that $e^{+j\omega t}$ has been used instead of $e^{-j\omega t}$. We note that

$e^{-j\omega t} = \cos\omega t - j\sin\omega t$, and

$$e^{+j\omega t} = \cos\omega t + j\sin\omega t \qquad (68)$$

This implies that our slightly altered Fourier transform of $x(t)$ is in fact 'looking for' the conjugate of $X(\omega)$ given by

$$X^*(\omega) = X_R(\omega) - jX_I(\omega)$$

Thus $\quad R_{xy}(\omega) = X^*(\omega) Y(\omega) \qquad (69)$

This means that we can use Fourier transforms to calculate cross correlation functions, as follows

Take two time domain records	$x(t)$	$y(t)$
Fourier transform	$X(\omega)$	$Y(\omega)$
Form the complex conjugate of either	$X^*(\omega)$	$Y(\omega)$
	or	
	$X(\omega)$	$Y^*(\omega)$

From the complex product $\quad R_{xy}(\omega) = X^*(\omega)Y(\omega)$

Inverse Fourier transform $\quad R_{xy}(\omega) \Rightarrow r_{xy}(\tau)$

This result has not been scaled to take account of the amplitudes of $x(t)$ and $y(t)$. In practice we would divide the result by the square root of the product of the variances of $x(t)$ and $y(t)$, that is

$$\sqrt{\sigma_x^2\,\sigma_y^2}\ .$$

7.3 Use of the FFT for Convolution in the Time Domain.

We have noted in the discussion in the previous section that we could use Fourier transforms to evaluate convolution and correlation integrals, and this method is most useful in the context of algebraic manipulation. However, we process our signals digitally and for time-to-frequency transformations we use the FFT. The question thus arises as to how we can use the FFT for these time domain operations. First, we have to remember that the FFT, notwithstanding its name, is in fact a Fourier *series* and implies periodicity in the time domain. This poses the problem illustrated on figure 22 in which record $x(n)$ is to be convolved with $h(n)$.

If $x(n)$ and $h(n)$ each represented one period of a periodic record then the convolution products at the right hand end of $x(n)$ will encroach on the next period; they would appear at the left hand side of the convolved result. This effect is known as *circularity* and such an operation is known as *circular convolution*. The practical way around this is to increase the data array lengths by inserting data points with zero values. Now, we recall that the FFT requires input data lengths which are an integer power of two, so the arrays must be increased further in length up to the next power of two. The operation is as follows:

Input data array $x(n)$	N points long
Input impulse response $h(n)$	M points long
Lengthen both arrays with zero valued data up to where I is an integer.	$L = 2^I \geq N + M - 1$
Take FFTs	$x(n) \Rightarrow X(k)$
	$h(n) \Rightarrow H(k)$
Complex multiply	$Y(k) = H(k)X(k)$
Inverse FFT	$Y(k) \Rightarrow y(n)$

$y(n)$ is the convolved result.

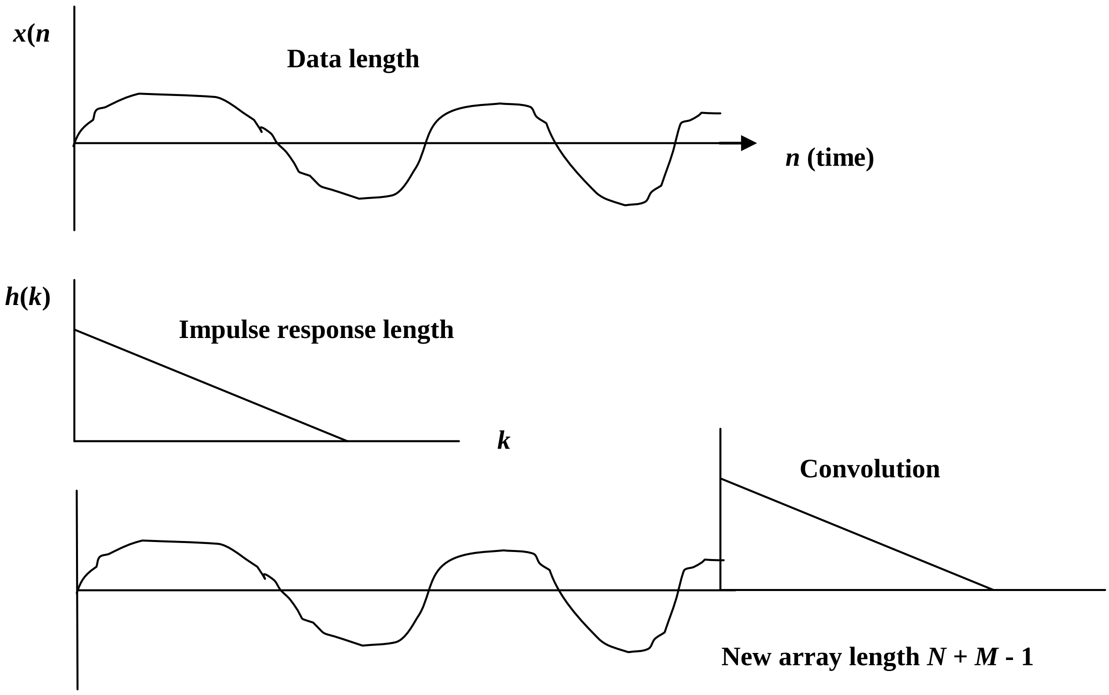


Figure 22. Record $x(n)$ to be convolved with $h(n)$, neither being periodic

7.4. Use of the FFT for Correlation in the Time Domain.

The same arguments apply here as they did for the convolution operation, and the process of using the FFT is similar, except that we need to conjugate one of the frequency domain records. It is

Input first data array $x(n)$	N points long
Input second data array y(n)	N points long
Double length of both arrays to	$L = 2N$ points
The right hand halves of both arrays are set to zero.	
Take FFTs	$x(n) \Rightarrow X(k)$
	$y(n) \Rightarrow Y(k)$
Conjugate one of them and complex multiply	$R_{xy}(k) = X^*(k)Y(k)$
Inverse FFT	$R_{xy}(k) \Rightarrow r_{xy}(n)$

The result is the unnormallised cross correlation function, which will be symmetrical about the origin of time. Because our data is periodic this implies that the components in negative time will appear at the right hand side of the array, figure 23.

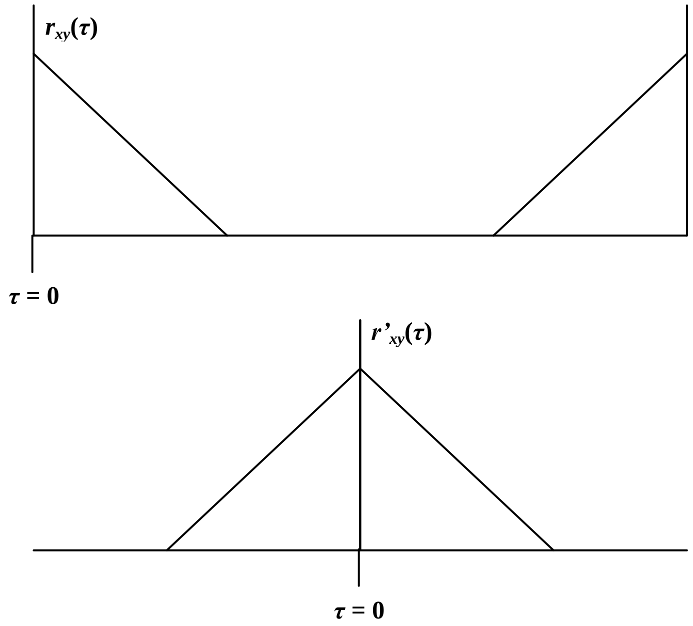


Figure 23. Cross-correlation function: (top) with negative lag components at the right hand side of the array, and (bottom) with a circular shift applied to set the origin of lag time in the centre of the array.

To get the CCF into the centre of the array we shift it in a circular fashion using the following operation: We consider a short sequence with L = 8; the unshifted array is $r_1(k)$ and the shifted array is $r_2(k)$.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8=*L***
$\boldsymbol{r_1(k)}$	$\boldsymbol{a}$	$\boldsymbol{b_1}$	$\boldsymbol{c_1}$	$\boldsymbol{d_1}$		$\boldsymbol{d_2}$	$\boldsymbol{c_2}$	$\boldsymbol{b_2}$
$\boldsymbol{r_2(k)}$		$\boldsymbol{d_2}$	$\boldsymbol{c_2}$	$\boldsymbol{b_2}$	$\boldsymbol{a}$	$\boldsymbol{b_1}$	$\boldsymbol{c_1}$	$\boldsymbol{d_1}$

To achieve this shift, set $r_2(1) = 0$ and $r_2(\frac{L}{2}+1) = r_1(1)$

Then, for $k = 2 \; to \; \frac{L}{2}$
set $r_2(\frac{L}{2}+k) = r_1(k)$
and $r_2(\frac{L}{2}-k+2) = r_1(L-k+2)$
End.

If the shift is to be done 'in place' using only a single array the operation is a little more complicated – we need to avoid overwriting our data. The operation requires a local variable P to store data temporarily – it is

Set $r(\frac{L}{2}+1) = r(1)$
Then, for $k = 2 \; to \; \frac{L}{2}$
set $P = r(\frac{L}{2}+k)$
$r(\frac{L}{2}+k) = r(k)$
$r(k) = P$
End.

8. Sampling and the Structure of Sampled Signals.

In Section 5 of this course we looked at symmetries in the DFS and showed that the frequency domain representation of a sampled signal repeated at intervals of the sampling frequency. This can be formally explained through the application of the theory of convolution – although this time the **convolution is in the frequency domain**. Consider the following convolution integral between two frequency domain functions

$$Y(\omega) = \int X(\omega - \Omega) S(\Omega)\, d\Omega \qquad (70)$$

$$\begin{aligned} y(t) &= \int Y(\omega) e^{j\omega t} d\omega \\ &= \int\int \left[X(\omega - \Omega) S(\Omega)\ d(\Omega) \right] e^{j\omega t} d\omega \\ &= \int\int \left[X(\omega - \Omega) e^{j(\omega - \Omega)t} d\omega \right] e^{j\Omega t} S(\Omega) d\Omega \end{aligned} \qquad (71)$$

There is a *frequency shift theorem* which is analogous to the *time shift theorem*. We work on the Fourier transform of the following function

$$\begin{aligned} \textbf{FT}\ [e^{j\Omega t} x(t)] &= \int x(t) e^{j\Omega t} e^{-j\omega t} dt \\ &= \int x(t) e^{-j(\omega - \Omega)t} dt \\ &= X(\omega - \Omega) \end{aligned}$$

Substituting this into equation 71 we get

$$\begin{aligned} y(t) &= \int \left[\int X(\omega - \Omega) e^{j(\omega - \Omega)t} d\omega \right] e^{j\Omega t} S(\Omega) d\Omega \\ &= x(t) \int S(\Omega) e^{j\Omega t} d\Omega, \end{aligned} \qquad (72)$$

so,

$$y(t) = x(t) s(t)\,. \qquad (73)$$

This shows that a multiplication of two signals in the time domain is equivalent to the convolution of their Fourier transforms. We now consider the sampling process as the multiplication of our signal by a series of impulse functions which repeat at the sampling frequency, figure 24. The operation is

$$y(t) = s(t) x(t)\,. \qquad (74)$$

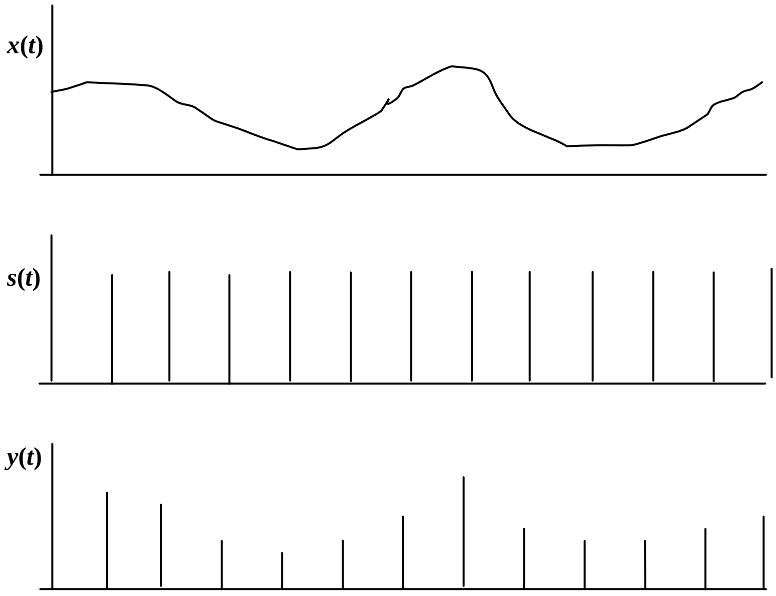


Figure 24. The sampling process where the continuous time signal $x(t)$ is multiplied by a series of impulse functions $s(t)$ which repeat at frequency f_s to produce the sampled signal $y(t)$.

The Fourier transform of $s(t)$ is obtained by considering the time shift theorem

If $\quad s(t) = \delta(t) + \delta(t-T) + \delta(t-2T) + \delta(t-3T) + \delta(t-4T)$.............. (75a)

Then $\quad S(\omega) = 1 + e^{-j\omega T} + e^{-2\omega T} + e^{-3\omega T} + e^{-4\omega T}$........... (75b)

where T is the sampling interval. This is a geometric progression, giving

$$S(\omega) = \frac{1}{1 - e^{-j\omega T}} \qquad (75c)$$

Now when $\omega T = 0, 2\pi, 4\pi$, etc, ω will be equal to integer multiples of the sampling frequency ω_s and $e^{j\omega T} = +1$. The function $S(\omega)$ rises to infinite values at these points – it can therefore be considered as a series of impulses spaced at intervals of the sampling frequency in the frequency domain. This is convolved with the spectrum of our continuous time signal, $x(t)$, and this causes the Fourier transform $X(\omega)$ to repeat at intervals of ω_s, figure 25. This repetition brings about the requirement to limit the bandwidth of an analogue signal that is to be digitised to half the sampling frequency. This requirement is incorporated in the *Nyquist Sampling Theorem*, restated here:

A continuous signal with frequency components in the range $f = 0$ to f_{max} can be reconstructed from a sequence of equally spaced samples, provided that the frequency of sampling, f_s, exceeds 2 f_{max}.

It is important to note here that the frequency components in the signal include the wanted signal **AS WELL AS** the noise. Before any sampling process we *band limit* our signals using lowpass filters, the *anti-aliasing filters*, discussed earlier.

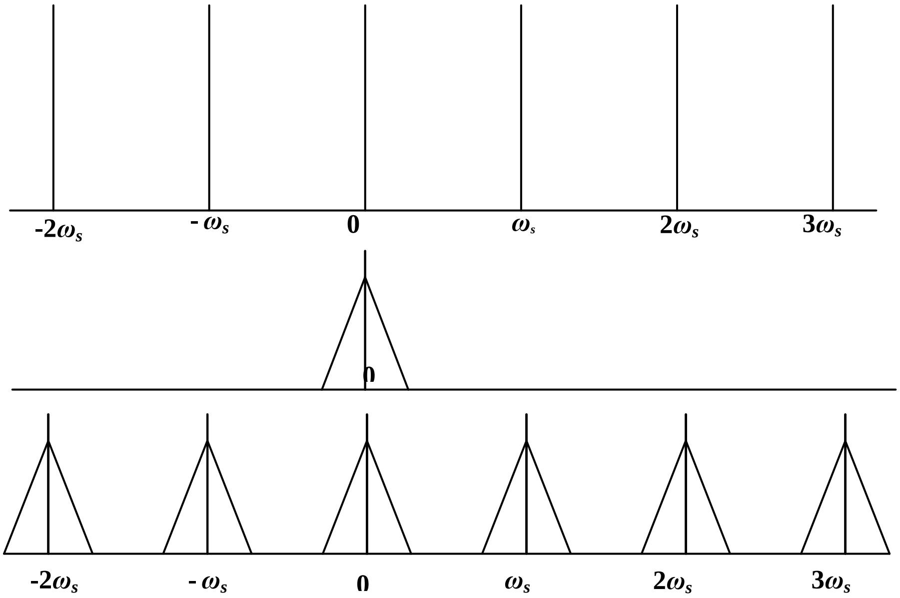


Figure 25. The sampling function $S(\omega)$ convolved with the spectrum of the input signal $X(\omega)$ to produce the spectrum of the sampled signal $Y(\omega)$ which is periodic in the frequency domain.

9. Signal Interpolation.

There are many situations where we would like to have more samples of a signal in a given time interval – that is, we wish to impart a higher sampling rate to the signal. **This will not provide any more information about the signal, nor will it make up for a sampling rate that was too low in the first place.** However it will provide a smoother (less jagged) signal for plotting on a graph, for example. The process of increasing the sampling rate is known as interpolation. It is equivalent to applying a rectangular window filter with limits $\pm\frac{\omega_s}{2}$ to the sampled data. An equivalent but far simpler operation is to take the FFT of the signal and then to double the length of both the real and imaginary arrays as illustrated below.

$x(n)$ $\quad x_1 \; x_2 \; x_3 \; x_4 \; x_5 \; x_6 \; x_7 \; x_8$

Take FFT

$R(k)$ $\quad R_1 \; R_2 \; R_3 \; R_4 \; 0 \; R_4 \; R_3 \; R_2$
$I(k)$ $\quad I_1 \; I_2 \; I_3 \; I_4 \; 0 \; -I_4 \; -I_3 \; -I_2$

Double array length and put zero values in the centre portion of both arrays.

$R(k)$ $\quad R_1 \; R_2 \; R_3 \; R_4 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; R_4 \; R_3 \; R_2$
$I(k)$ $\quad I_1 \; I_2 \; I_3 \; I_4 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; -I_4 \; -I_3 \; -I_2$

Take inverse FFT

$x(n)$ $\quad x_1 \; x_1' \; x_2 \; x_2' \; x_3 \; x_3' \; x_4 \; x_4' \; x_5 \; x_5' \; x_6 \; x_6' \; x_7 \; x_7' \; x_8 \; x_8'$

The x'_n values are new samples, interpolated between the original ones. It will be clear that the operation can only increase the number of samples by integer powers of two. The operation is as follows.

Original data, N points in array	$x(n)$
Interpolated data $L = MN$ points in array	$x'(n)$
Take FFT of	$x(n) \Rightarrow X(k) = X_R(k) + jX_I(k)$
Set up receiving arrays L points long	$X'(k) = X'_R(k) + jX'_I(k)$

For $k = 2 \; to \; \frac{N}{2}$
$X'_R(L + k - \frac{N}{2}) = X_R(k + \frac{N}{2})$
$X'_I(L + k - \frac{N}{2}) = X_I(k + \frac{N}{2})$
$X'_R(k) = X_R(k)$
$X'_I(k) = X_I(k)$
End.

For $k = (\frac{N}{2} + 1) \; to \; (L - \frac{N}{2} + 1)$
$X'_R(k) = 0$
$X'_I(k) = 0$
End.

$X'_R(1) = X_R(1)$
$X'_I(1) = X_I(1)$ (76)
End.

10. Introduction to the Laplace Transform and Filter Responses.

The Laplace transform is a frequency domain transformation much along the lines of the Fourier transform. It can be regarded as a more comprehensive operation than the Fourier transform in that it breaks down a time domain waveform not only in terms of individual frequencies but also in terms of whether the amplitudes of these frequency components are growing or diminishing as time progresses. We can imagine a three dimensional plot with frequency on the vertical axis, growth or decay on the horizontal axis, and amplitude on the third axis, perpendicular to the paper; such a plot is known as the *s*-plane, figure 26.

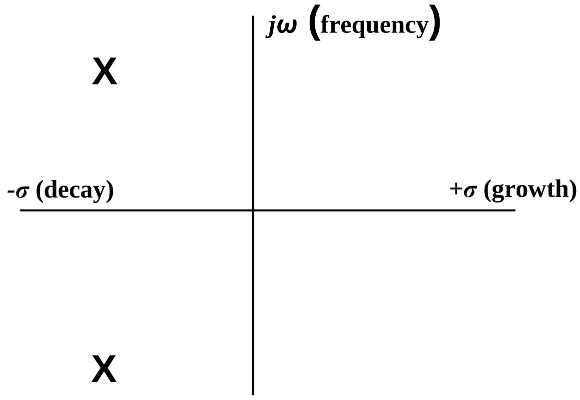


Figure 26. *s*-plane plot.

Imagine a very much idealised violin string that emits a single frequency when plucked. We could plot the decaying sound as a pair of crosses on figure 26 – one for the positive frequency, and one for the negative frequency to take account of the frequency domain symmetry which follows from symmetries in the Fourier transform. A decaying or growing elemental sinusoid can be expressed in the time domain as

$$x(t) = e^{j\omega t} e^{\sigma t} \tag{77}$$

Where σ is the decay/growth rate. The Laplace transform does this in shorthand:

$$x(t) = e^{st}, \tag{78a}$$

where $s = \sigma + j\omega$ (78b)

s is known as *complex frequency*, or the *Laplace variable*. The Laplace transform invokes a correlation process (multiply and integrate) to seek out matches to e^{st} for particular values of $s = \sigma + j\omega$. It is expressed as

$$X(s) = \int_0^\infty x(t) e^{-st}\, dt \tag{79}$$

.

Unlike the Fourier transform the integral is single sided in that the integration starts at $t = 0$ instead of $t = -\infty$. Laplace transforms are used to describe linear systems such as filters in just the same way as the Fourier transform. Figure 27 shows the inputs and outputs of a filter in terms of the time domain, the real frequency domain (Fourier), and the complex frequency domain (Laplace). The output of the filter can be obtained by convolution in the time domain – the input signal with the impulse response $h(t)$, multiplication in the real frequency domain – Fourier transform of the input multiplied by the frequency response, or multiplication in the complex frequency domain – multiplication of the Laplace transform of the input by the Laplace transform of the impulse response. All of the theorems that apply to the Fourier transform (such as superposition, linearity, timeshift, frequency shift, differentiation and integration) also apply to the Laplace transform. We use these theorems to describe the responses of electric circuits in Laplace transform terms. For example the simple resistance-capacitance lowpass filter shown on figure 15 has the following input output relationship in the time domain:

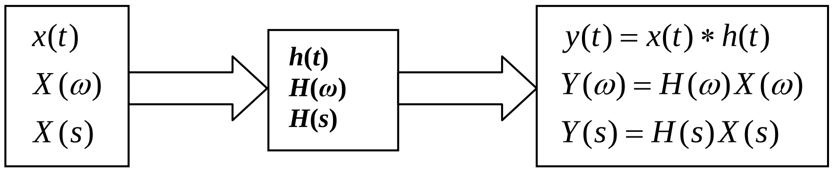


Figure 27. System responses in the time domain, as Fourier transforms, and as Laplace transforms.

$$y(t) = \frac{1}{C}\int i\,dt \quad \text{or} \qquad i = C\frac{dy}{dt} \tag{80a}$$

Now $$i = \frac{x(t) - y(t)}{R} \tag{80b}$$

Combining these two equations we get

$$y(t) + CR\frac{dy}{dt} = x(t) \tag{80c}$$

In a similar manner to the Fourier transform, in the case of the Laplace transform the operation of differentiation is achieved by multiplication by *s*. We can then transform the whole equation

$$Y(s) + sCR\,Y(s) = X(s) \tag{81}$$

CR is the circuit time constant and is also the inverse of the cut-off frequency of the filter. We set

$$\omega_0 = \frac{1}{CR} \qquad \text{and obtain}$$

$$H(s) = \frac{Y(s)}{X(s)} = \frac{\omega_0}{\omega_0 + s} \tag{82}$$

H(*s*) is in fact the Laplace transform of the circuit impulse response, given by

$$H(s) = \int_0^\infty h(t)e^{-st}\,dt \tag{83}$$

H(*s*) is known either as the *complex frequency response* or as the *transfer function* of the filter. We can plot *H*(*s*) on our two-dimensional *s*-plane plot of figure 26, see figure 28. The cross on the diagram is known as a *pole*; it has an infinite value when *s* is wholly real and equal to ω_0. We can imagine the modulus value of *H*(*s*) as a surface on the *s*-plane, coming out of the paper. Contours and the equivalent surface on the s-plane are shown on figure 29.

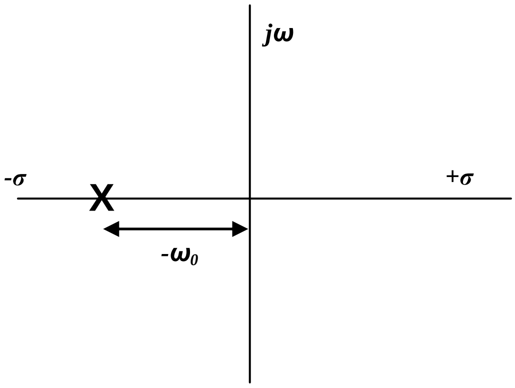


Figure 28. s-plane plot for first order RC lowpass filter.

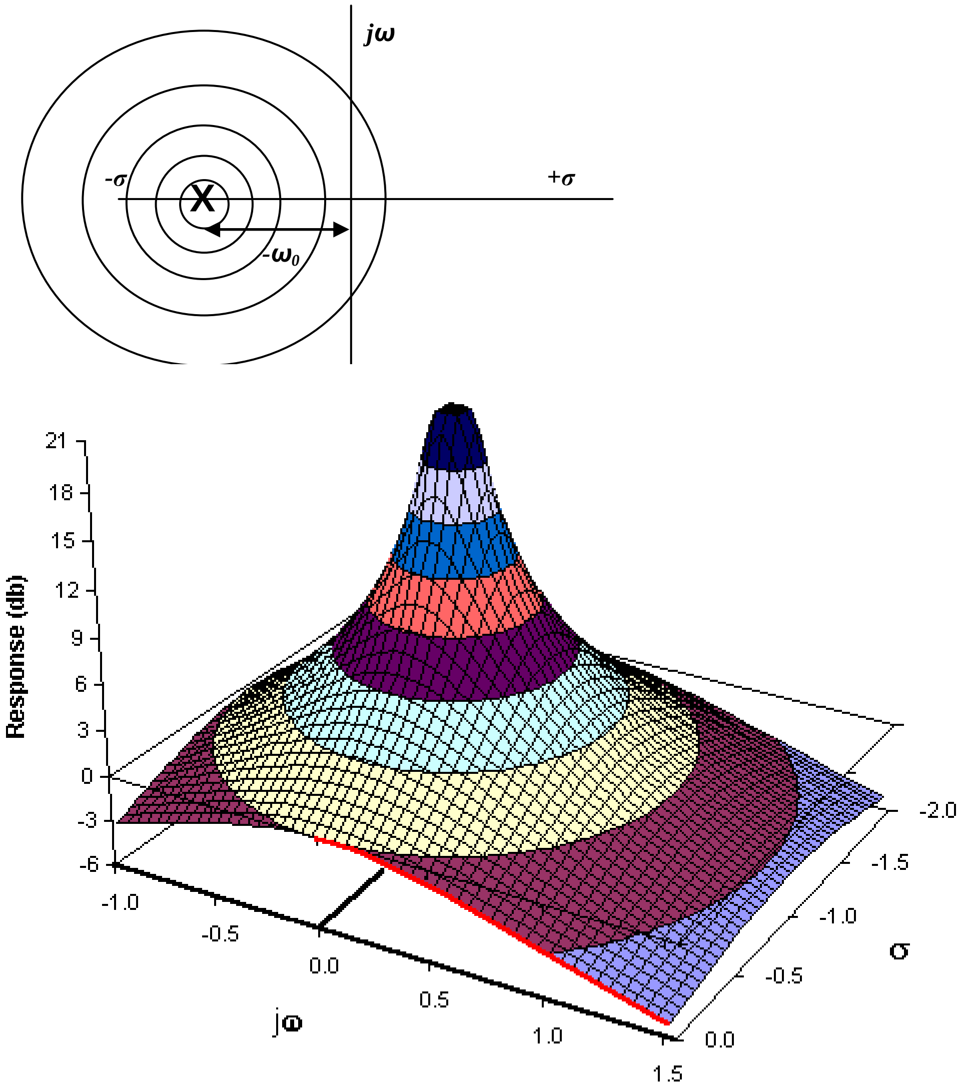


Figure 29. (Top) contours of $H(s)$ on the s-plane, and (bottom) modulus of $H(s)$ as a surface on the s-plane.

We are looking down on a volcano like structure with an infinitely tall pole in its centre. We can imagine a slice through the foothills of our volcano along the line of the $j\omega$ axis. Viewing this slice from the right hand side of the diagram we would see the shape of figure 30. This shape is in fact the frequency response of the filter. What we have done here is evaluated $H(s)$ for $s = j\omega$; that is

$$|H(s)|_{s=j\omega} = |H(\omega)| = \left|\frac{\omega_0}{\omega_0 + j\omega}\right| \tag{84}$$

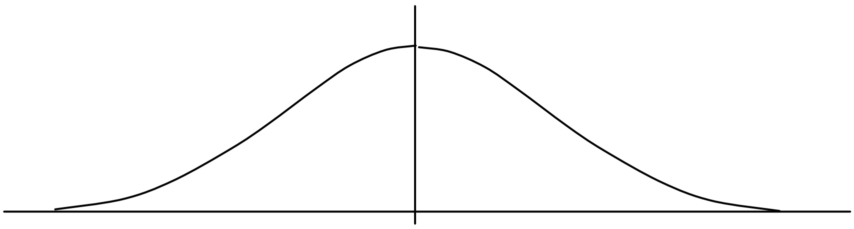


Figure 30. View towards the vertical axis of a slice through the s-plane – the frequency response of the filter.

This way of looking at responses can be extended to more complex filters. For example a resistance-inductance-capacitance low pass filter has the transfer function

$$H(s) = \frac{\omega_0^2}{s^2 + \dfrac{\omega_0 s}{q} + \omega_0^2} \tag{85}$$

The denominator has roots at

$$s = \frac{-\dfrac{\omega_0}{q} \pm \sqrt{\dfrac{\omega_0^2}{q^2} - 4\omega_0^2}}{2} \quad = \quad -\frac{\omega_0}{2q} \pm j\frac{\omega_0}{2q}\sqrt{4q^2 - 1} \tag{86}$$

We can plot these roots on our s-plane, figure 31.

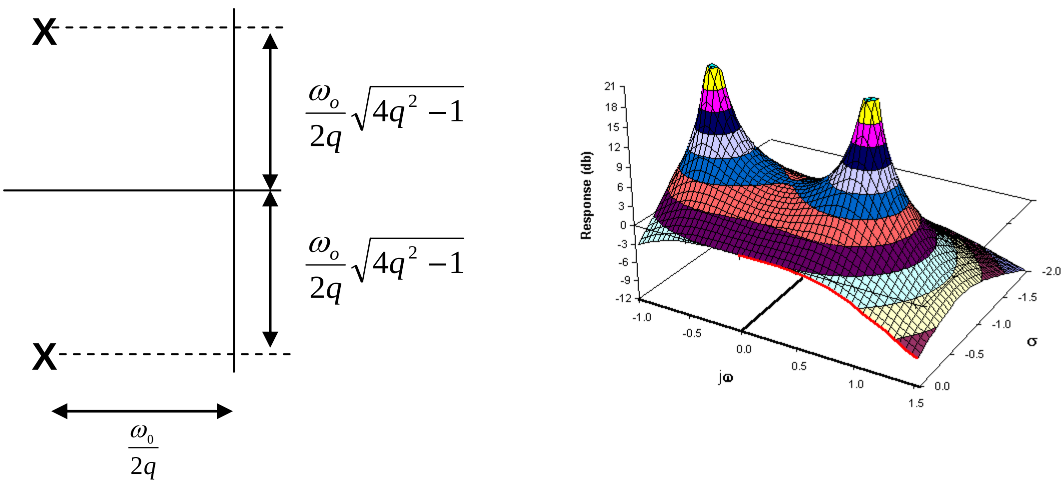


Figure 31. (left) s-plane poles of the second order lowpass filter, and (right) the corresponding modulus surface.

10.1. Filter Responses in More Detail.

The two-pole filter described above is known as a *second order section.* It will cut off at a rate of 40dB per decade, or 12dB per octave – why? To answer this we evaluate the transfer function on the $j\omega$ axis.

$$H(\omega) = \frac{\omega_0^2}{\omega_0^2 - \omega^2 + j\dfrac{\omega_0 \omega}{q}} \qquad (87)$$

At low frequency ($\omega \to 0$) $H(\omega)$approximates to unity and there is no signal loss. At high frequency ($\omega \rangle\rangle \omega_0$) and

$$H(\omega) \approx \frac{\omega_0^2}{\omega^2}; \qquad (88)$$

the cut off is proportional to the square of frequency. If ω doubles then $H(\omega)$ is divided by 4 and the cut off is

$$20\log_{10}\frac{1}{4} \quad = \quad -12 \text{ dB per octave.}$$

If ω is multiplied by ten then $H(\omega)$ is divided by 100 and the cut off is

$$20\log_{10}\frac{1}{100} \quad = \quad -40 \text{ dB per decade.}$$

In practice we may require a sharper cut-off rate than this, so we would cascade sections in series, two second order sections giving a cut off rate of 80 dB per decade, and so on upwards. We may also include a first order section in our filter chain giving an extra 20 dB per decade to our filter system. As an aid to design we illustrate filters by means of a simple block diagrams, each block representing a first or second order section in terms of its order, pole frequency ω_0 and q, figure 32.

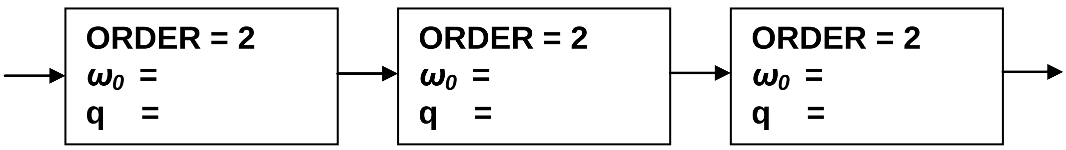


Figure 32. Block diagram of a filter chain; each block is specified by its order, its pole frequency and its *q*.

It is very important to note that the overall cut-off frequency of the filter will not be the same as any of the values of ω_0 ; indeed, for high order filters many of the section values of ω_0 are very different from the overall cut off frequency. The table below gives scaled values of ω_0 and second order section q's for four types of fourth order filter; the values of ω_0 are scaled, so we see the lengths of the *pole vectors* on the s-plane relative to each other.

Filter Type	ω_0	***q***	***phase***
Butterworth	1.0	1.31	68
	1.0	0.54	22
Chebyschev (0.5dB)	1.03	2.94	80
	0.60	0.71	45
Chebyschev (3.0dB)	0.95	5.58	85
	0.44	1.08	62
Bessel-Thomson	3.02	0.52	16
	3.39	0.81	52

The most commonly used of these filter types is the Butterworth design which has a response in its pass band which is *maximally flat*. The pole positions for third order versions of these filters are illustrated on figure 33, and the pole positions for various orders of Butterworth filter are shown on figure 34.

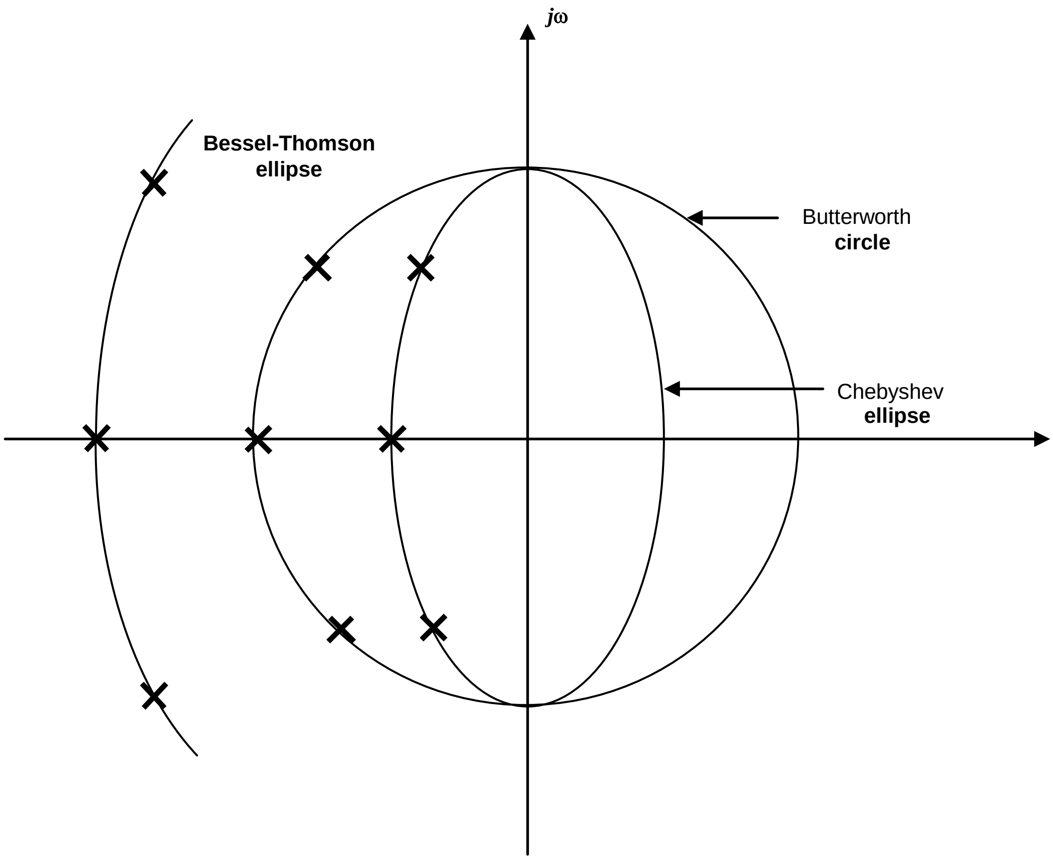


Figure 33. Pole positions for the Butterworth, Chebyschev and Bessel-Thomson filters.

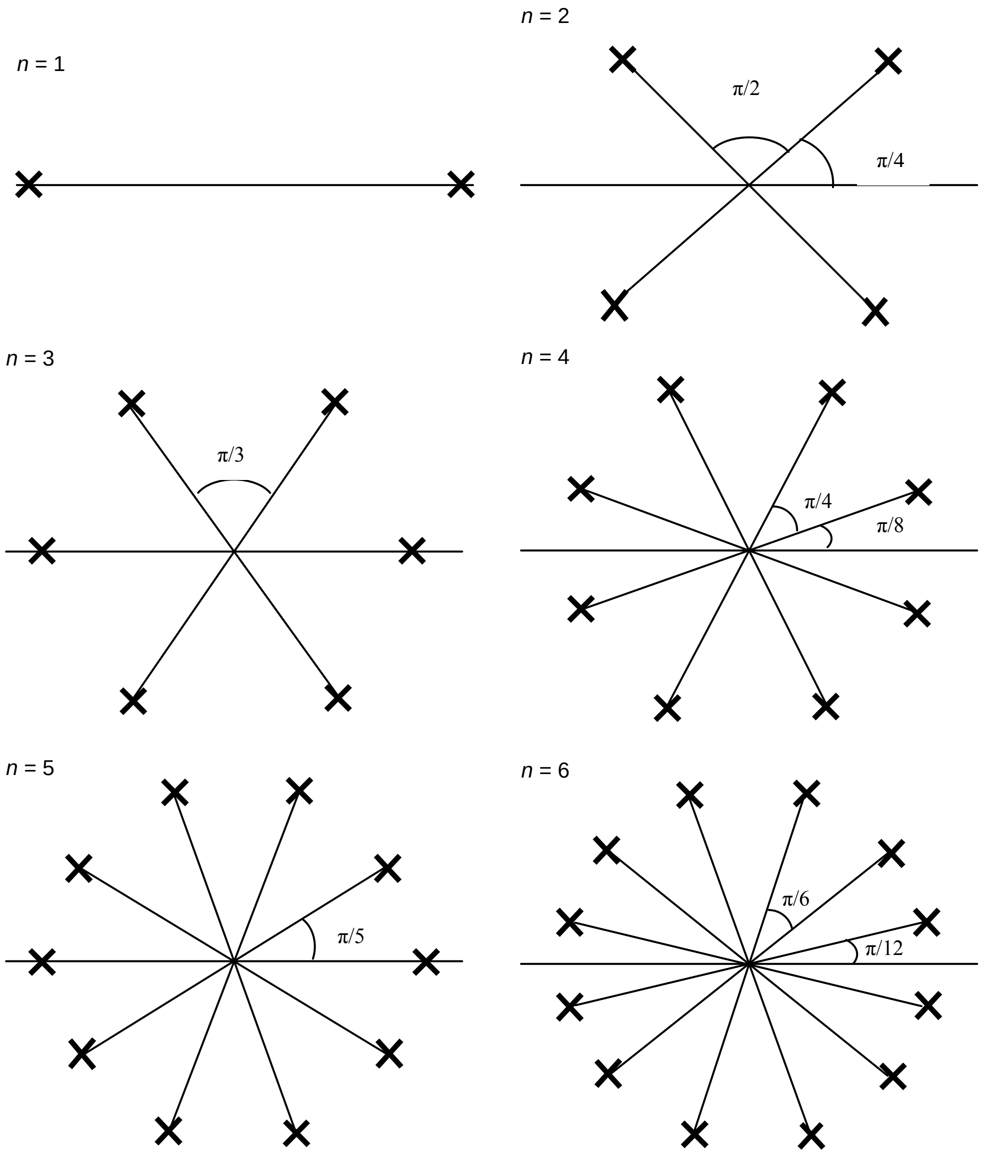


Figure 34. Pole positions for Butterworth filters of various orders.

The Chebyschev filter has an initial cut off rate which is sharper than the Butterworth design but this is achieved at the expense of ripples in the frequency response below the cut off frequency – that is, ripples in the pass band. The Bessel-Thomson design has a very gentle transition into its cut off band in order to minimise *phase distortion*, discussion of which is beyond the scope of this short course.

10.2 Generalised Filter Responses.

So far we have only considered lowpass filters – that is, those that pass low frequencies and attenuate high frequencies. However, in the general case we may require filters that attenuate frequency components in a particular band, or alternatively filters that accentuate components in a frequency band. Figure 35 shows a graphical way of illustrating the specification of a filter. As before, the horizontal axis represents frequency; the vertical axis is calibrated in terms of the attenuation that the filter will apply to the signal usually expressed in dBs. The ideal specification is represented by the heavy solid lines, showing two *stop bands* and a single *pass band* between them. This type of response cannot be achieved in practice; infinitely sharp cut-offs would, if they were achievable, be associated with instability in the time domain responses – spurious oscillation would arise at signal transients, and these would mask important features in our signals. Sharp cut-offs are also associated with ripple in the pass and stop bands of filters. Figure 35 also illustrates the compromise between a high rate of cut-off required by the 'ideal' specification and ripples in the pass and stop bands.

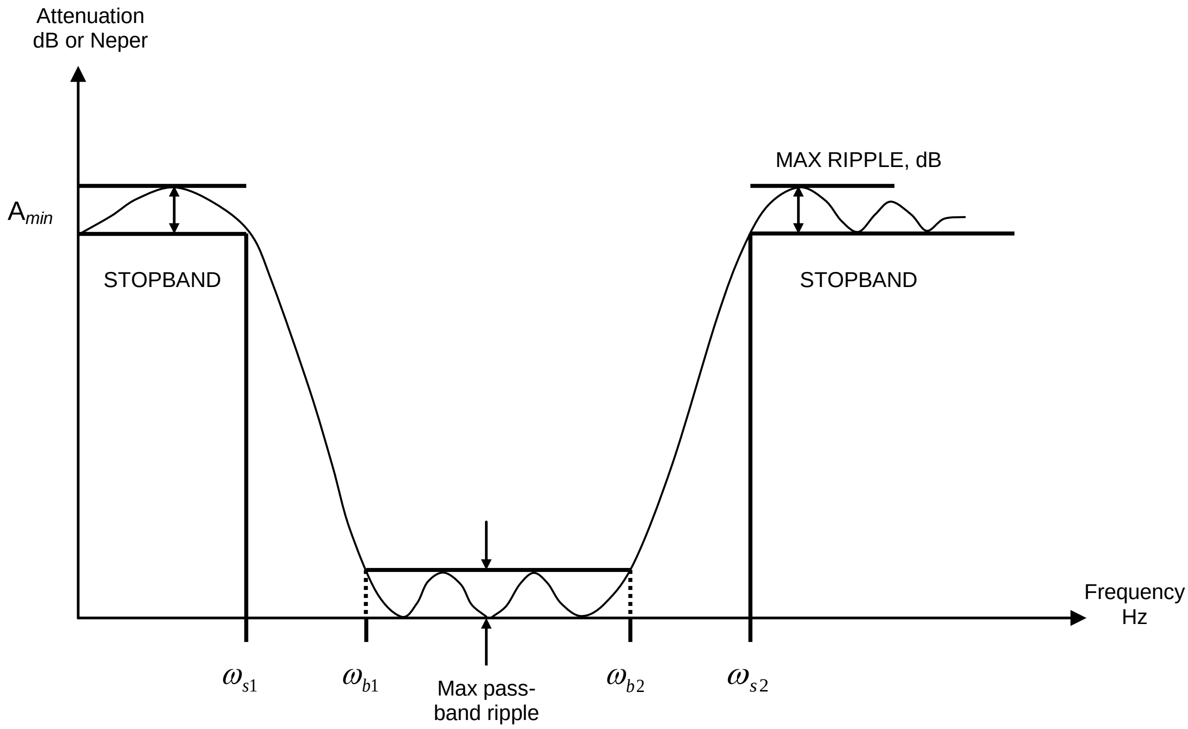


Figure 35. Specification of a filter showing the ideal responses and the practically achievable compromise.

What might be specified is the maximum ripple allowed, expressed in dB as well as a number of corner frequencies – on figure 35 ω_{s1} *and* ω_{s2} define the stop band edges whilst ω_{b1} *and* ω_{b2} define the limits of the pass band.

The algebraic descriptions of filters of the type of complexity shown in figure 35 is beyond the scope of this course, although they broadly follow the applications of Laplace transform calculus that we have seen so far for simple lowpass designs, although the methods are extended to include points of **zero response** on the *s*-plane. Zeros correspond to a zero factor in the numerator of the transfer function, for example

$$H(s) = \frac{(s+\omega_1)(s+\omega_2)}{(s+\omega_3)(s+\omega_4)} \tag{89}$$

is a transfer function which as poles at s = $-\omega_3$ and $s = -\omega_4$, and zeros at s = $-\omega_1$ and $s = -\omega_2$. It is illustrated on figure 36.

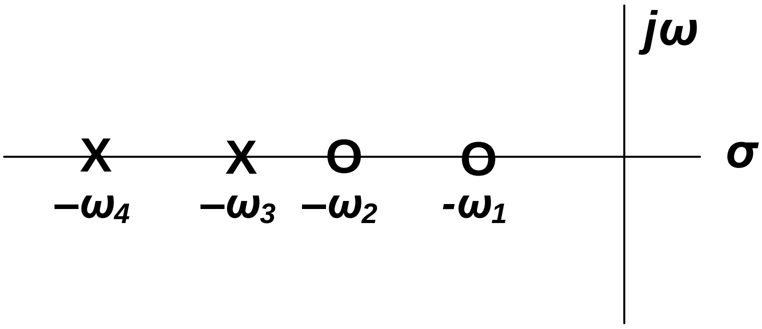


Figure 36. Filter with poles at s = $-\omega_3$ and $s = -\omega_4$, and zeros at s = $-\omega_1$ and $s = -\omega_2$.

11. Analogue Hardware Filters.

In the previous section we saw how filters of high order are built up from basic blocks of first and second order transfer functions. There are many ways in which these basic blocks can be implemented. We have already seen in section 5.6 of this course how they could be implemented using the FFT - the transfer function is evaluated on the $j\omega$ axis by setting $s = j\omega$ and the resulting frequency response is then mapped into the real and imaginary arrays of the FFT, forming a window by which the 'signal' FFT is multiplied to get the filter output. Alternative ways of filtering digitally will be discussed in connection with z-transform calculus in section 12. It is the purpose of this section to introduce students to the way filters may be implemented using analogue electronic hardware. In many senses this could be regarded as a 'pre-digital' and rather historic technique, although it is important to note that in many applications hardware filters are very much in use – anti-aliassing, for example. Analogue filters can be divided into two categories *passive*, and *active*.

11.1. Passive Filters.

These are constructed from passive electronic components – resistors, capacitors and inductors. The most common first order building blocks are shown on figure 37.

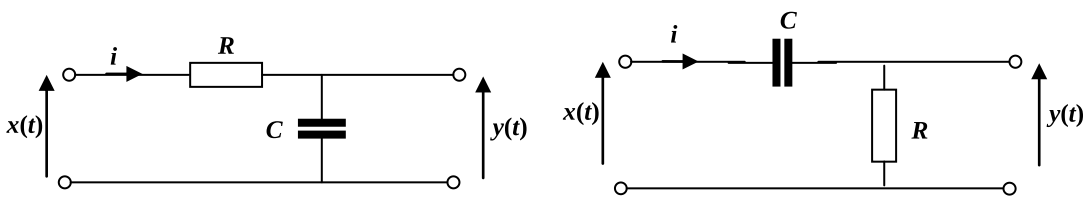


Figure 37. Passive first order lowpass (left), and highpass (right) filters.

The transfer functions are:

$$H(s) = \frac{\omega_0}{s + \omega_0} \qquad \text{lowpass,}$$

and

$$H(s) = \frac{s}{s + \omega_0} \tag{90}$$

where $\omega_0 = \frac{1}{CR}$ in both cases.

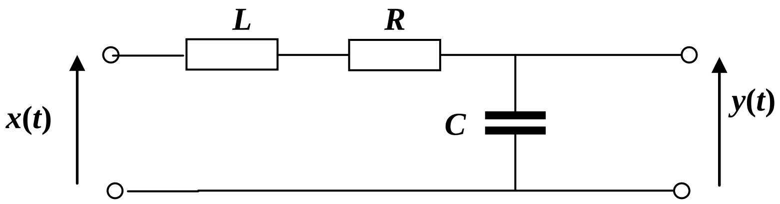


Figure 38. Second order lowpass passive filter.

The basic second order section for a lowpass filter is shown on figure 38; it has the transfer function given earlier in equation 85, namely

$$H(s) = \frac{\omega_0^2}{s^2 + \frac{\omega_0 s}{q} + \omega_0^2},$$

where $\omega_0 = \sqrt{1/LC}$, $q = \omega_0 L / R$, and $\frac{\omega_0}{q} = \frac{R}{L}$

There are many reasons for avoiding the use of inductances, the principal being that they are both heavy and bulky. Also, passive filters cannot be *simply* cascaded to give filters of higher order, the reason being that a first or second order section in the filter chain will load the section before and will be loaded by the section after it, changing the transfer function. It is possible to use a buffer amplifier between stages to minimise loading effects but there are enormous advantages to be had in using the amplifiers themselves, wired with appropriate feedback to form the filter element; such systems are known as active filters.

11.2. Active Filters – Introduction.

Active filters avoid the use of inductances and can easily be cascaded to form high order designs without the disadvantage of loading between sections. There are many different possible circuit configurations and in this course we will describe the most common – the Sallen and Key second order lowpass section. Their principle of operation is the use of positive and negative feedback to change the way in which passive elements form the circuit frequency response. These feedback effects can be understood by consideration of the *Miller effect*. Figure 39 shows an amplifier of gain A with a feedback impedance Z connected between output and input.

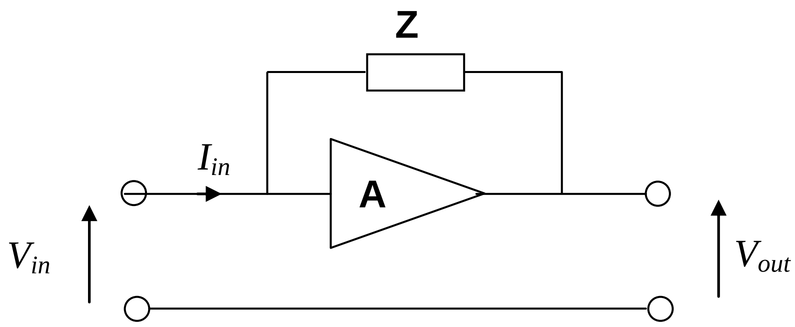


Figure 39. Illustration of the Miller effect.

We consider the input impedance of the circuit

$$Z_{in} = \frac{V_{in}}{I_{in}} \tag{91a}$$

The input current is

$$I_{in} = \frac{V_{in} - V_{out}}{Z}, \tag{91b}$$

and $V_{out} = AV_{in}$, whence

$$I_{in} = \frac{V_{in}(1 - A)}{Z} \tag{91c}$$

The input impedance is then

$$Z_{in} = \frac{Z}{1 - A} \tag{91d}$$

If A = 1 then Z_{in} is infinite. If A negative then Z_{in} is less than Z, but of the same sign. If A is positive and greater than unity the denominator becomes negative and the sign of Z is changed when it is reflected into Z_{in}. If Z were formed of a capacitor the input impedance would be

$$Z_{in} = \frac{j}{\omega C|(1 - A)|} \tag{92}$$

Z_{in} would thus offer a phase response similar to an inductor. Capacitors are used in this way in active filters, the most common of which is the Sallen and Key design, which we describe next.

11.3. The Sallen and Key Lowpass Filter.

The circuit is shown in figure 40; its algebraic analysis is rather long and tedious, so will not be included here.

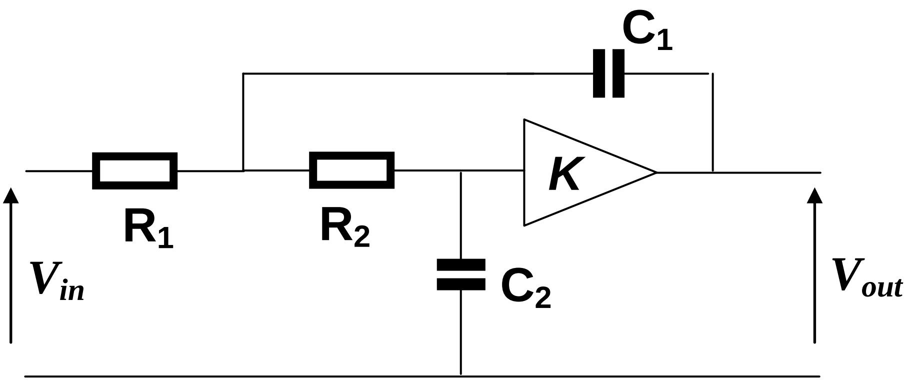


Figure 40. The Sallen and Key lowpass second order section.

The principal results are

$$\omega_0^2 = \frac{1}{R_1 R_2 C_1 C_2} \tag{93a}$$

$$q = \frac{\sqrt{\dfrac{1}{R_1 R_2 C_1 C_2}}}{\dfrac{1}{R_1 C_1} + \dfrac{1}{R_2 C_1} + \dfrac{1-K}{R_2 C_2}}, \tag{93b}$$

And, as before

$$H(s) = \frac{\omega_0^2}{s^2 + \dfrac{\omega_0 s}{q} + \omega_0^2}. \tag{93c}$$

The question now arises as to how we might design the circuit, given such a wide choice of variables - two resistances, two capacitances, and the amplifier gain *K*. Luckily, a much simplified and systematic process is available. We start with the so-called *normalised circuit* shown on figure 41; this has a pole frequency $\omega_0 = 1$, a q specified by its two capacitors, and input resistances of value 1 Ohm. These circuit values are in no way practicable, but are for a step in the design process which involves *de-normalisation*, as follows.

- Scale up R_1 and R_2 to values *R*, typically 10 k to 1 MΩ.
- Scale down C_1 and C_2 by factor *R*; this maintains the value of *q* but keeps $\omega_0 = 1$.
- Scale down C_1 and C_2 by factor equal to the required pole frequency ω_0.

The component values that result are

$$R_1 = R_2 = R$$
$$C_1 = \frac{2q}{\omega_0 R} \qquad (94)$$
$$C_2 = \frac{1}{2q\omega_0 R}$$

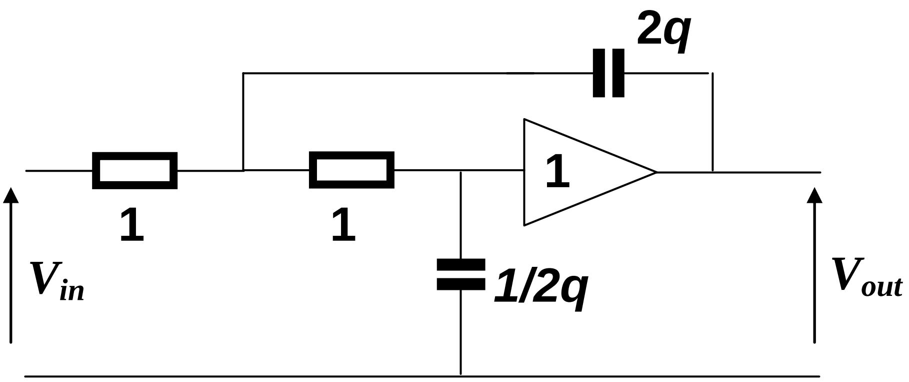


Figure 41. Normallised Sallen and Key circuit.

12. Digital Filters and the z-Transform.

In this section we consider digital filters that are implemented in the time domain as distinct from using the FFT and associated windows in the frequency domain. The theoretical basis of these filters lies in the *z-transform* which is a frequency domain transformation, and as we will see later, is not unrelated to the Fourier and Laplace transforms. The transform variable *z* is defined as

$$z = e^{sT}$$

Where *s* is complex frequency and *T* is the interval between samples of a digitised signal; $T = 1/f_s$ where f_s is the sampling frequency. Recalling the shift theorems, we note that *z* corresponds to a shift of time *T*. There are many ways to conceive of, and indeed to design, digital filters and we will deal with just a few in this course. By way of introduction we will consider a discrete form a simple first order electric filter. Some properties of the *z*-transform will then be presented and these will be followed by a brief discussion of the design of simple digital filters that are not based on hardware analogue filters.

12.1. Digital Implementations of Electric Filters.

We return to our basic first order lowpass filter, figure 15. We have already seen the differential equation that describes this filter in the time domain (equation 80c), repeated here

$$y(t) + CR\frac{dy(t)}{dt} = x(t) \qquad (95a)$$

Imagine now that we have our signals in digital form; we can write the equation thus

$$y(n) + CR\frac{y(n) - y(n-1)}{T} = x(n) \qquad (95b)$$

The second term on the left hand side is the backward difference which mimics the operation of differentiation. We can rearrange this equation to get

$$y(n) = \frac{x(n) + \frac{CR}{T}y(n-1)}{\left(1 + \frac{CR}{T}\right)} \qquad (95c)$$

This is a *recurrence relationship* that relates our output data to our input data. If we had a signal recorded into array $x(n)$ and an output array $y(n)$ that begins with all values set to zero we could implement this filter on array $x(n)$. We would set $y(1)$ to

$$y(1) = \frac{x(1)}{\left(1 + \frac{CR}{T}\right)}$$

Then from $y(2)$ onwards we would calculate $y(n)$ using equation 95c, taking input sample data $x(n)$ and the previously calculated $y(n - 1)$. The term CR/T is important - recall that

$T = \dfrac{1}{f_s} = \dfrac{2\pi}{\omega_s}$ and that $CR = \dfrac{1}{\omega_0}$, then

$$\frac{CR}{T} = \frac{\omega_s}{2\pi\omega_0}$$

This implies that the operation of our digital filter does not take account of either the sampling rate or the actual cut off frequency of our electric filter; they key variable is the ratio $\dfrac{\omega_0}{\omega_s}$ - the cut off frequency divided by the sampling frequency.

12.2. A Better Differentiator.

The digital filter derived above is based on the simplest form of discrete differentiator, the backward difference. It is only a good representation of the differentiation operation in situations where the data sampling rate is high in relation to the rate at which signals might change during their passage through the system, and in the case of our simple electric filter this implies that ω_s is very much greater than ω_0. In many practical situations our data is not sampled at such a high rate and so we consider a new differentiation scheme that is more precise under these circumstances. Figure 42 shows two samples of a digitised waveform $x(n)$. We can set the backward difference to be equal to the average of the differential coefficient of $x(n)$ at n and (n-1).

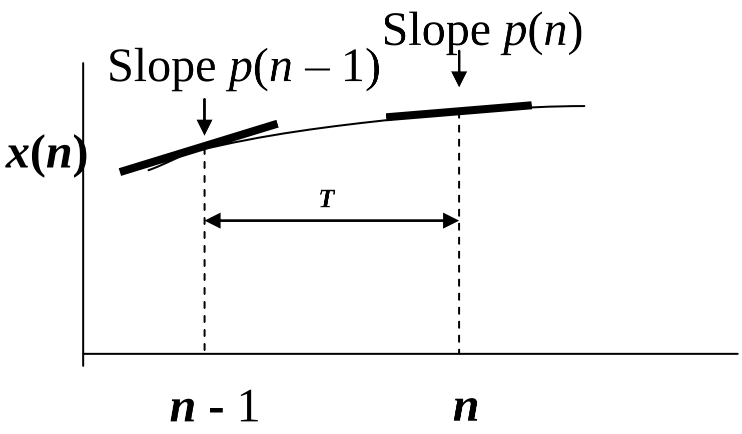


Figure 42. An improved differentiation scheme.

Let the differentiated function in continuous time be

$$p(t) = \frac{dx(t)}{dt} \qquad (96a)$$

Expressed in discrete time our differentiation operation is

$$\frac{p(n) + p(n-1)}{2} = \frac{x(n) - x(n-1)}{T} \qquad (96b)$$

The differentiated function $p(n)$ can be obtained from $x(n)$ by the following recurrence relationship

$$p(n) = \frac{2}{T}\left[x(n) - x(n-1)\right] - p(n-1) \qquad (96c)$$

Clearly, this recurrence relationship could be incorporated into the discretisation process for equation 95a, or indeed for differential equations for filters of higher order. This step in digital filter design is much simplified if we use the z-transform.

12.3 The *z*-transform.

The z-transform is a frequency domain transform which derives from the Laplace transform in the context of sampled signals. It is essentially a shorthand notation which incorporates the time delay of one sample period between successive samples; figure 43 shows a short series of samples of different weights.

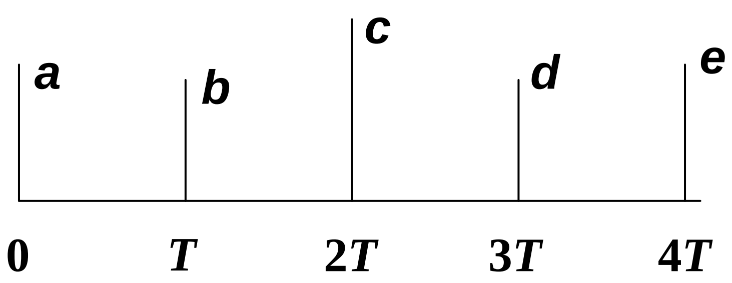


Figure 43. A series of time domain samples of different weights.

In the time domain this can be expressed as a sequence of appropriately weighted impulse functions

$$x(t) = a\delta(t) + b\delta(t-T) + c\delta(t-2T) + d\delta(t-3T) + e\delta(t-4T) \tag{97a}$$

The Laplace transform of this sequence is

$$X(s) = a + be^{-sT} + ce^{-2sT} + de^{-3sT} + ee^{-4sT} \tag{97b}$$

The corresponding z-transform is

$$X(z) = a + bz^{-1} + cz^{-2} + dz^{-3} + ez^{-4} \tag{97c}$$

We now look at how the z-transform representation can deal with differentiation in the time domain. Our simple differentiator, the backward difference, can be written as

$$p(t) = \frac{y(t_n) - y(t_{n-1})}{T} \tag{98a}$$

Its z-transform is

$$P(z) = \frac{Y(z) - z^{-1}Y(z)}{T} \quad = \quad \frac{z-1}{zT}Y(z) \tag{98b}$$

Returning to equation 81 for the Laplace transform of our lowpass filter we merely substitute the above expression wherever there is a multiplication by the Laplace variable s, since this represents differentiation in the time domain. We can thus write the z-transform directly

$$Y(z) + CR\frac{z-1}{zT}Y(z) = X(z) \tag{99}$$

Rearranging we get

$$H(z) = \frac{Y(z)}{X(z)} = \frac{zT}{(T+CR)z - CR} \tag{100a}$$

This z-transform represents our filter. In fact, this is how digital filters are most commonly described, although the majority are more complicated. The question we need to answer now is how to get from the z-transform back to the discrete time domain in order that we can operate on our sampled data. This is done by the following steps:

Step 1: Multiply the numerator and the denominator by an appropriate power of z which makes all terms have a zero or negative power of z. In the current case we divide numerator and denominator by z to get

$$H(z) = \frac{T}{(T + CR) - CRz^{-1}} \tag{100b}$$

Step 2: Now set $Y(z) = H(z)X(z)$

or $$Y(z) = \frac{T}{(T + CR) - CRz^{-1}} X(z)$$

Step 3: Cross multiply and rearrange

$$Y(z)(T + CR) = TX(z) + CRz^{-1}Y(z)$$

Step 4: By appropriate division remove the multiplier to the $Y(z)$ term

$$Y(z) = \frac{T}{T + CR} X(z) + \frac{CR}{T + CR} z^{-1}Y(z)$$

Step 5: Return to the discrete time domain – by inspection

$$y(n) = \frac{x(n)}{1 + \frac{CR}{T}} + \frac{\frac{CR}{T}}{1 + \frac{CR}{T}} y(n-1) \tag{100c}$$

This corresponds exactly to equation 95c which we derived earlier. It is the recurrence relationship that implements our filter on input data $x(n)$ to get output data $y(n)$.

12.4 Implementation Using the Improved Differentiator.

To implement our filter with the improved differentiator we follow exactly the same procedure except that we substitute a different expression for the Laplace variable s to represent differentiation in the time domain. In the time domain our improved differentiator is

$$\frac{1}{2}\left[p(n) + p(n-1)\right] = \frac{1}{T}\left[x(n) - x(n-1)\right] \tag{101a}$$

This corresponds to the following substitution for s

$$s \Rightarrow \frac{2}{T}\frac{z-1}{z+1} \tag{101b}$$

This is known as the *bilinear transformation*. To use it on our filter we transform equation 81 to get

$$Y(z) + \frac{2CR}{T}\frac{z-1}{z+1}Y(z) = X(z) \tag{102a}$$

Following the steps given above we rearrange this expression to get

$$Y(z) = \left[\frac{1}{1+\frac{2CR}{T}}\right]\left[X(z) + z^{-1}X(z)\right] - \left[\frac{1-\frac{2CR}{T}}{1+\frac{2CR}{T}}\right]z^{-1}Y(z) \tag{102b}$$

Before deriving the recurrence relationship we consider the multiplying factors

$$CR = \frac{1}{\omega_0} \qquad \text{and} \qquad T = \frac{2\pi}{\omega_s} \qquad \text{whence} \qquad \frac{2CR}{T} = \frac{\omega_s}{\pi\omega_0}$$

Thus $\dfrac{1}{1+\dfrac{2CR}{T}} = \dfrac{\dfrac{\pi\omega_0}{\omega_s}}{1+\dfrac{\pi\omega_0}{\omega_s}}$ and $\dfrac{1-\dfrac{2CR}{T}}{1+\dfrac{2CR}{T}} = \dfrac{\dfrac{\pi\omega_0}{\omega_s}-1}{\dfrac{\pi\omega_0}{\omega_s}+1}$

By inspection our recurrence relationship becomes

$$y(n) = a\left[x(n) + x(n-1)\right] - by(n-1)\,, \tag{103}$$

where

$$a = \frac{\dfrac{\pi\omega_0}{\omega_s}}{1+\dfrac{\pi\omega_0}{\omega_s}} \qquad \text{and} \qquad b = \frac{\dfrac{\pi\omega_0}{\omega_s}-1}{\dfrac{\pi\omega_0}{\omega_s}+1}$$

We note again that the filter coefficients are expressed in terms of the quotient of the filter cut off frequency ω_0 and the sampling frequency ω_s.

12.5 The *z*-Transform More Formally.

The z-transform of the sequence $x(n)$ is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{104}$$

We have defined z as the complex number

$$z = e^{sT} = e^{\sigma T} e^{j\omega T} \tag{105}$$

z can be mapped on to a complex plane known as the z-plane, figure 44.

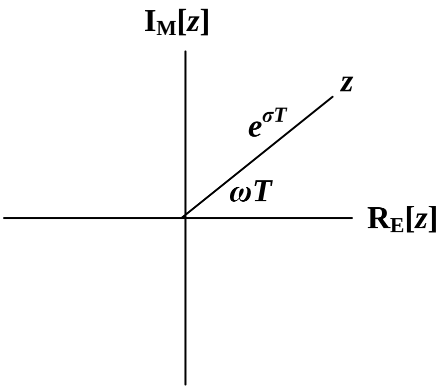


Figure 44. The z-plane.

z is a vector of length $e^{\sigma T}$ which forms an angle ωT radians with respect to the real axis of the z-plane. We can map features on the s-plane on to the z-plane, figure 45.

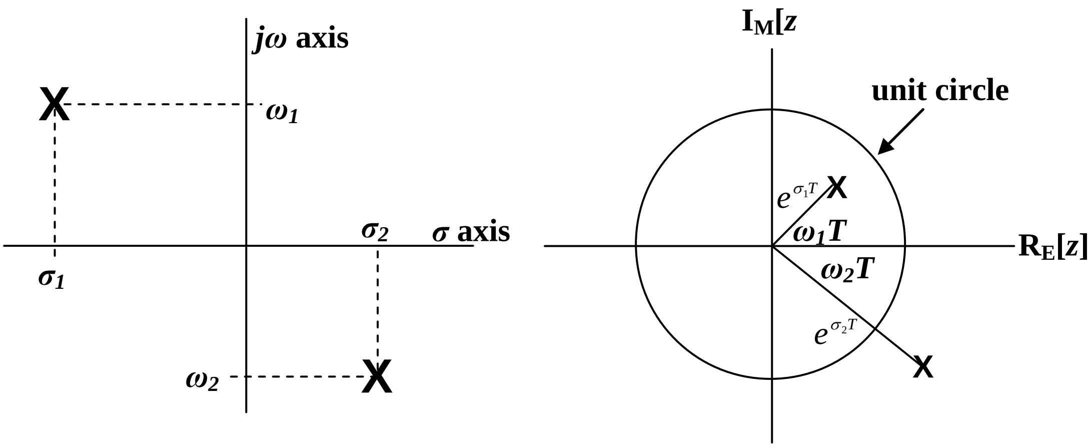


Figure 45. Mapping features on the *s*-plane on to the *z*-plane.

The $j\omega$ axis of the s-plane corresponds to $\sigma = 0$ and $e^{\sigma T} = 1$, that is, $z = e^{j\omega T}$. This implies that the image of the $j\omega$ axis of the s-plane is a circle of unit radius on the z-plane. If we imagine the modulus value of the z-transform, $|X(z)|$, as a surface on the z-plane, then the height of the function above the z-plane on the unit circle corresponds to the modulus Fourier transform of $x(n)$. A very important feature of the z-plane is the position of key frequencies such as the sample frequency ω_s; we note that

$N\omega_s T = N2\pi$,

So the point z = 1 corresponds to a frequency of zero, ω_s , and integer multiples of ω_s, figure 46.

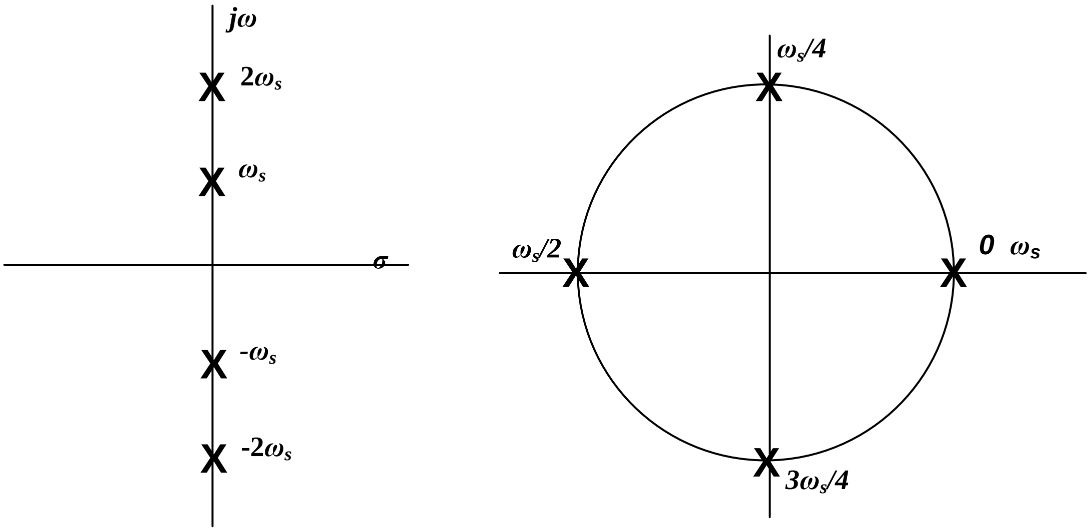


Figure 46. Periodicity on the $j\omega$ axis of the s-plane mapping on to the unit circle of the z-plane.

Other key frequencies map as follows:

$$\frac{\omega_s}{4} \Rightarrow z = e^{j\frac{\pi}{2}} \qquad \frac{\omega_s}{2} \Rightarrow z = e^{j\pi} \qquad \frac{3\omega_s}{4} \Rightarrow z = e^{j\frac{3\pi}{4}}$$

Thus we see that the successive repeats at multiples of ω_s for frequency domain functions of sampled signals all overlay each other on the z-plane – very convenient and very efficient!

We can express this idea algebraically

$$X(z)_{z=e^{j\omega T}} = \sum_{n=-\infty}^{\infty} x(n)e^{j\omega nT} \qquad (106)$$

Now, expressing frequency as a fraction of the sampling frequency, $\omega = \frac{k}{N}\omega_s$, we get

$$X(z)_{z=e^{j\omega T}} = \sum_{n=-\infty}^{\infty} x(n)e^{jnk\omega_s T/N} = \sum_{n-0}^{N-1} x(n)e^{-jnk\frac{2\pi}{N}} \qquad (107)$$

This proves that $X(z)$ evaluated on the unit circle is the discrete Fourier transform of $x(n)$.

12.6 *z*-Plane Descriptions of Analogue Filters.

To arrive at the *z*-plane description of our filters we need to determine the values of *z* which correspond to zero values and infinite values of the transfer functions – the zeros and the poles. Our lowpass filter with simple backward difference differentiation has the transfer function

$$H(z) = \frac{zT}{(T + CR)z - CR} \tag{108}$$

This has a zero at $z = 0$ and a pole at

$$z = \frac{CR}{T + CR} \quad = \quad \frac{1}{1 + \dfrac{2\pi\omega_0}{\omega_s}}$$

Figure 47 shows the corresponding *z*-plane plot. Imagining $|H(z)|$ as a surface on the *z*-plane we find that the filter has a maximum pass at $z = 1$, ($\omega = 0$), and that the response falls away as we approach $z = -1$, ($\omega = \frac{\omega_s}{2}$). For our filter with the bilinear differentiator we have

$$H(z) = \frac{z + 1}{z\left(1 + \dfrac{2CR}{T}\right) - \left(\dfrac{2CR}{T} - 1\right)} \tag{109}$$

This has a zero at z = -1, ($\omega = \frac{\omega_s}{2}$), which stops any possible aliasing, as well as a pole at

$$z = \frac{1 - \dfrac{\pi\omega_0}{\omega_s}}{1 + \dfrac{\pi\omega_0}{\omega_s}}$$

Figure 47 also shows the *z*-plane plot for this filter; note that the zero and pole positions are different from those for the filter with the backward difference differentiator.

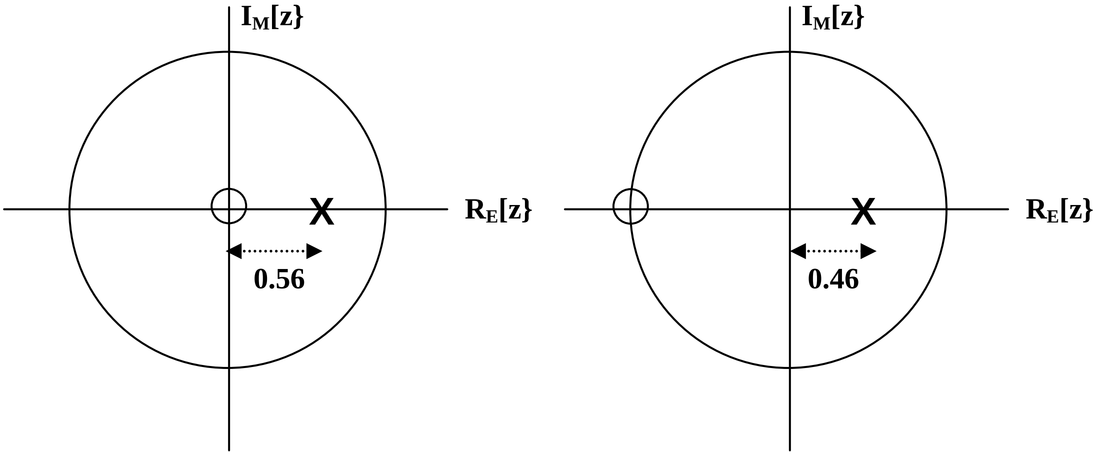


Figure 47. *z*-plane plots for simple lowpass filter with cut off frequency $\omega = \omega_s / 8$. (Left) with backward difference differentiation, and (right) for differentiation using the bilinear transformation.

12.7 The *z*-Transform and Convolution in the Time Domain.

We related the input and output of our filters through the *z*-domain transfer function $H(z)$. Now $H(z)$ behaves for sampled signals in exactly the same way as $H(s)$ or $H(\omega)$ behave for continuous time signals. It can be proved that the multiplication of *z*-transforms is equivalent to convolution in the time domain. So, if $h(t)$ is the impulse response of our filter, convolution in the time domain is

$$y(t) = \int_{-\infty}^{\infty} x(t-\tau)h(\tau)d\tau \,, \qquad (110a)$$

$$Y(z) = H(z)X(z) \,, \qquad (110b)$$

and

$$\begin{aligned} Y(z) &\Leftrightarrow y(t) \\ X(z) &\Leftrightarrow x(t) \\ H(z) &\Leftrightarrow h(t) \end{aligned} \qquad (110c)$$

are transform pairs, just as was the case for Fourier and Laplace transform descriptions of our filter.

13. Digital Filters with no Electrical Counterparts.

There are many types of digital filter which are not based on electric circuit prototypes – indeed, processing of signals in digitised form has given considerable freedom in how filters are designed. Here we consider a few basic designs which illustrate the essential principles, although it is to be noted that the range of practical possibilities is enormous.

13.1 The Moving Average Filter.

The operation of the moving average filter is shown on figure 48. The input sequence is $x(n)$ and the output sequence is $y(n)$. Each output is calculated by adding N inputs, in this case $N = 4$.

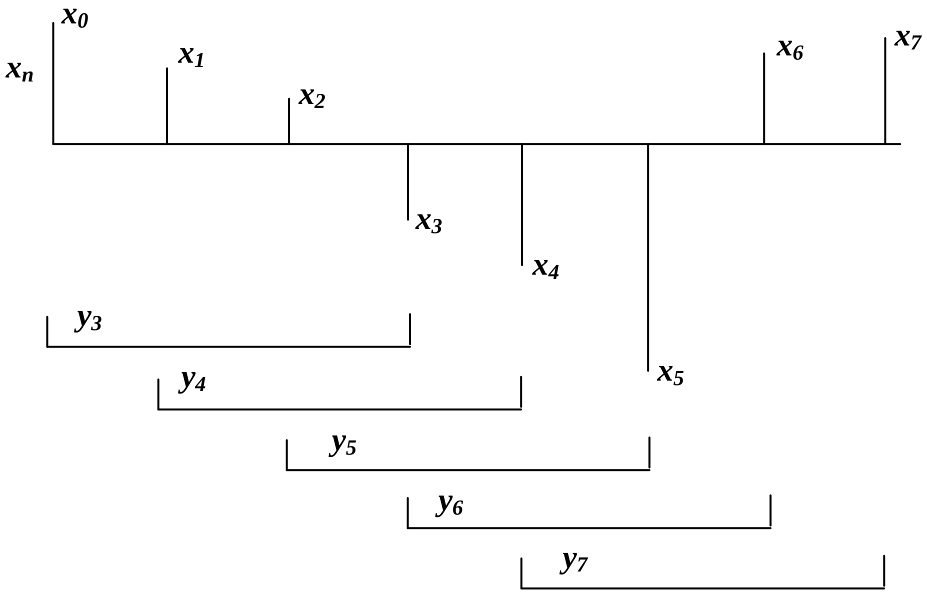


Figure 48. Operation of the moving average filter.

Calculating successive outputs we have

$$y(3) = x(0) + x(1) + x(2) + x(3) \qquad (111a)$$

$$y(4) = \qquad x(1) + x(2) + x(3) + x(4) \qquad (111b)$$

$$y(5) = \qquad\qquad x(2) + x(3) + x(4) + x(5) \qquad (111c)$$

We note that there are terms common to successive output samples, for example

$$y(5) = y(4) - x(1) + x(5), \text{ and} \qquad (111d)$$
$$y(4) = y(3) - x(0) + x(4)\,. \qquad (111e)$$

This corresponds to the following recurrence relationship

$$y(n) = y(n-1) + x(n) - x(n-N) \qquad (112)$$

For discrete signals we conceive of the impulse response as being a unit sample response – that is, the response of our filter to a data sample of unit magnitude. If $x(n)$ consisted of just a single sample at $n = 0$ then the response of our filter would be a short series of N unit samples, figure 49.

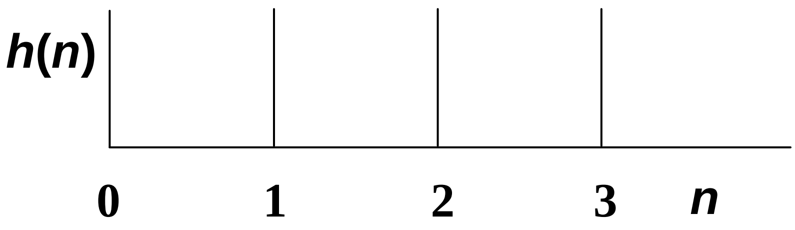


Figure 49. Unit sample response for 4-point moving average.

This has the z-transform

$$H(z) = 1 + z^{-1} + z^{-2} + z^{-3}, \qquad (112a)$$

Which is a geometric progression with the sum

$$H(z) = \sum_{n=0}^{N} z^{-n} = \frac{1 - z^{-N}}{1 - z^{-1}} = \frac{z^N - 1}{(z-1)z^{N-1}} \qquad (112b)$$

We are now going to do two things:

- We are going to illustrate the derivation of the recurrence relationship from the z-transform.
- We will map $H(z)$ on to the z-plane.

To get the recurrence relationship we go back a step and rearrange $H(z)$ into negative powers of z.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - z^{-N}}{1 - z^{-1}}$$

Cross-multiplying we get

$$Y(z) = z^{-1}Y(z) + X(z) - z^{-N}X(z) \qquad (112c)$$

The recurrence relationship is found by inspection

$$y(n) = y(n-1) + x(n) - x(n-N) \qquad (112d)$$

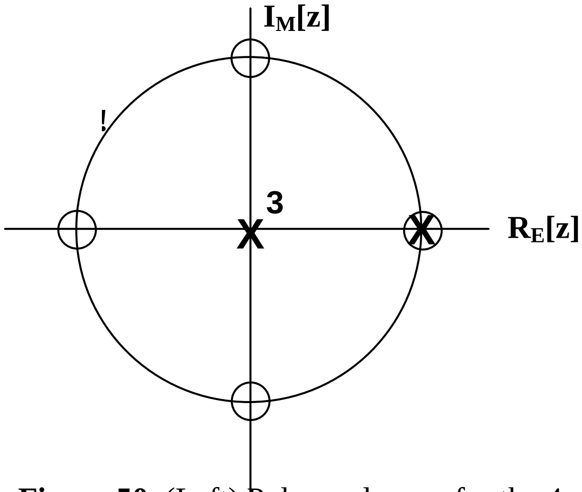
This is identical to the expression derived earlier.

To map $H(z)$ on to the z-plane we seek factors in the numerator and denominator of $H(z)$. The numerator is zero when

$$z^N = 1 = e^{j2\pi m},$$

where m is an integer. So zeros will fall at $z = e^{j2\pi\frac{m}{N}}$. They will appear on the z-plane unit circle at intervals of $\frac{2\pi}{N}$; for N = 4 this corresponds to z = 1, +j, -1, -j, figure 50.

The poles of $H(z)$ correspond to zero factors in the denominator – there will be (N - 1) poles at z = 0, and a single pole at z = 1, cancelling the zero there and defining the centre of the pass band. The corresponding frequency response is also shown on figure 50.



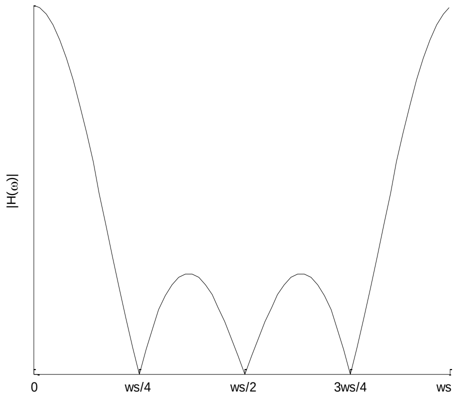


Figure 50. (Left) Poles and zeros for the 4-point moving average filter, and (right) the corresponding frequency response.

As it stands, this filter will have gain at zero frequency, which we derive by evaluating $H(z)$ in the limit as z tends towards unity.

$$H(1) = lt_{z\to 1}\left[H(z)\right] \quad = \quad lt_{\delta\to 0}\left[\frac{(1+\delta)^N - 1}{(1+\delta-1)(1+\delta)^{N-1}}\right] \quad = \quad \frac{N\delta}{\delta} \quad = \quad N \tag{113}$$

If a filter gain of unity is required then all of the input terms are divided by N.

13.2 Simple Bandpass and Highpass filters.

The pole cancelling the zero at z = 1 imparted the lowpass characteristic to the filter just described. We can use this principle as a basis for designing filters which are bandpass or highpass. For a simple filter with a pass band at $\omega_S/4$ we draw the z-plane plot shown on figure 51. We use the same total number of poles as zeros since this prevents impractical delays or advances in the recurrence relationship. The zeros at $z = \pm 1$ cause total attenuation at $\omega = 0$ and at $\omega = \omega_S/2$. Our filter will pass at $\omega = \pm\omega_s / 4$.

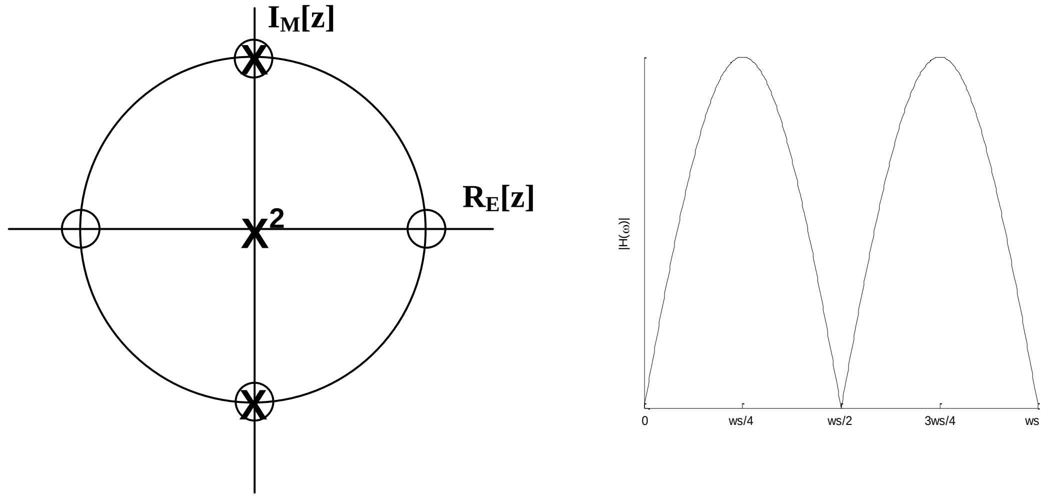


Figure 51. *z*-plane plot and frequency response for simple bandpass filter.

By inspection of the *z*-plane we write down the *z*-transform

$$H(z) = \frac{z^4 - 1}{(z-j)(z+j)z^2} = \frac{z^4 - 1}{(z^2+1)z^2} = \frac{1 - z^{-4}}{1 + z^{-2}} = \frac{Y(z)}{X(z)} \qquad (114a)$$

So, $$Y(z) = -z^{-2}Y(z) + X(z) - z^{-4}X(z), \qquad (114b)$$

From which we obtain the time domain recurrence relationship by inspection

$$y(n) = -y(n-2) + x(n) - x(n-4) \qquad (114c)$$

Following similar principles the *z*-plane plot for a simple highpass design and the resulting frequency response are shown on figure 52.

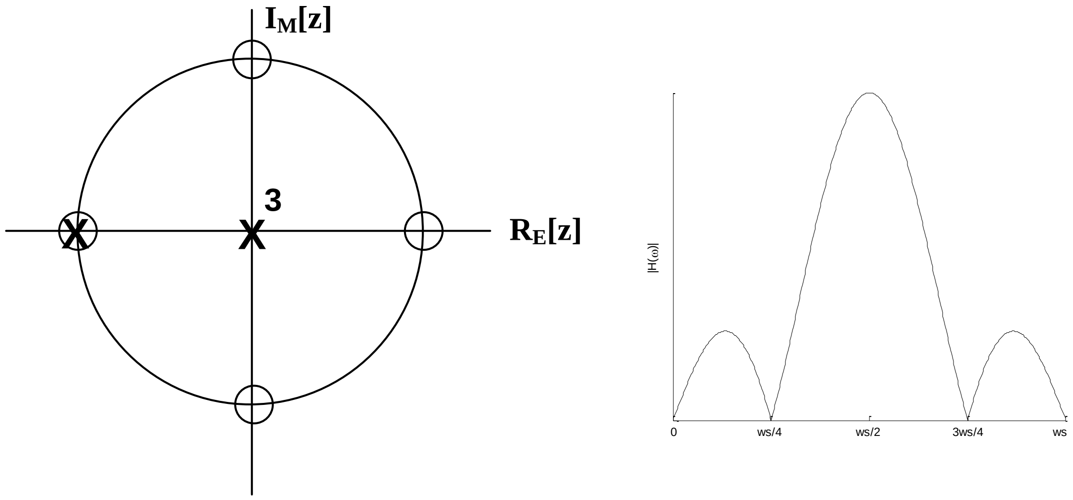


Figure 52. *z*-plane plot and frequency response for simple highpass filter.

Here we have

$$H(z) = \frac{z^4 - 1}{(z+1)z^3} = \frac{1 - z^{-4}}{1 + z^{-1}}, \tag{115a}$$

from which follows the recurrence relationship

$$y(n) = -y(n-1) + x(n) - x(n-4) \tag{115b}$$

13.3 Analytic Derivation of Impulse Responses from the z-Transform.

In most practical situations the impulse responses of digital filters are obtained by actually running them on a computer, and we shall see how to do this in an exercise. For completeness in our discussion we will consider here how the impulse response can be obtained by algebraic derivation. The method is to manipulate $H(z)$ to a form which is a power series in z^{-1}, thus.

Lowpass case

$$H(z) = \frac{z^N - 1}{(z-1)z^{N-1}} = 1 + z^{-1} + z^{-2} + z^{-3} ....z^{-(N-1)}, \tag{116a}$$

whence the impulse response is

$$h(n) = \delta(0) + \delta(1) + \delta(2) + \delta(3) + ...\delta(N-1) \tag{116b}$$

Bandpass case

$$H(z) = \frac{z^4 - 1}{(z^2+1)z^2} = 1 - z^{-2}, \tag{117a}$$

whence

$$h(n) = 1 - \delta(2) \tag{117b}$$

Highpass case

$$H(z) = \frac{z^4 - 1}{(z+1)z^3} = 1 - z^{-1} + z^{-2} - z^{-3}, \tag{118a}$$

and

$$h(n) = \delta(0) - \delta(1) + \delta(2) - \delta(3) \tag{118b}$$

These impulse responses are shown of figure 53.

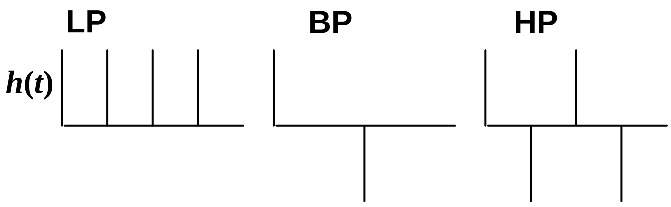


Figure 53. Impulse responses for the simple lowpass, bandpass and highpass filters.

13.4 A Simple Notch Filter.

To achieve a notch filter we place a pair of zeros on the *z*-plane unit circle at a position corresponding to the frequency ω_0 at which we require the notch, figure 54. ω_0 corresponds to the *z*-plane vector angles $\theta = \pm\frac{\omega_0}{\omega_s}2\pi$. We then place poles on the same vectors, close to the unit circle – these control the width of the notch in frequency space. By inspection we obtain the *z*-transform

$$H(z) = \frac{\left(z - e^{j\theta}\right)\left(z - e^{-j\theta}\right)}{\left(z - pe^{j\theta}\right)\left(z - pe^{-j\theta}\right)} = \frac{z^2 - 2z\cos\theta + 1}{z^2 - 2pz\cos\theta + p^2} = \frac{1 - 2z^{-1}\cos\theta + z^{-2}}{1 - 2pz^{-1}\cos\theta + p^2 z^{-2}} = \frac{Y(z)}{X(z)} \qquad (119a)$$

So

$$Y(z) = 2p\cos\theta z^{-1}Y(z) - p^2 z^{-2}Y(z) + X(z) - 2\cos\theta z^{-1}X(z) + z^{-2}X(z) \qquad (119b)$$

The recurrence relationship follows by inspection

$$y(n) = b_1 y(n-1) + b_2 y(n-2) + x(n) + a_1 x(n-1) + x(n-2)\,, \qquad (119c)$$

where $a_1 = -2\cos\theta \qquad b_1 = 2p\cos\theta \qquad b_2 = -p^2$.

Three frequency responses for this filter are shown on figure 54 for $\omega_0 = \omega_s/8$ and p = 0.5, 0.7 and 0.9.

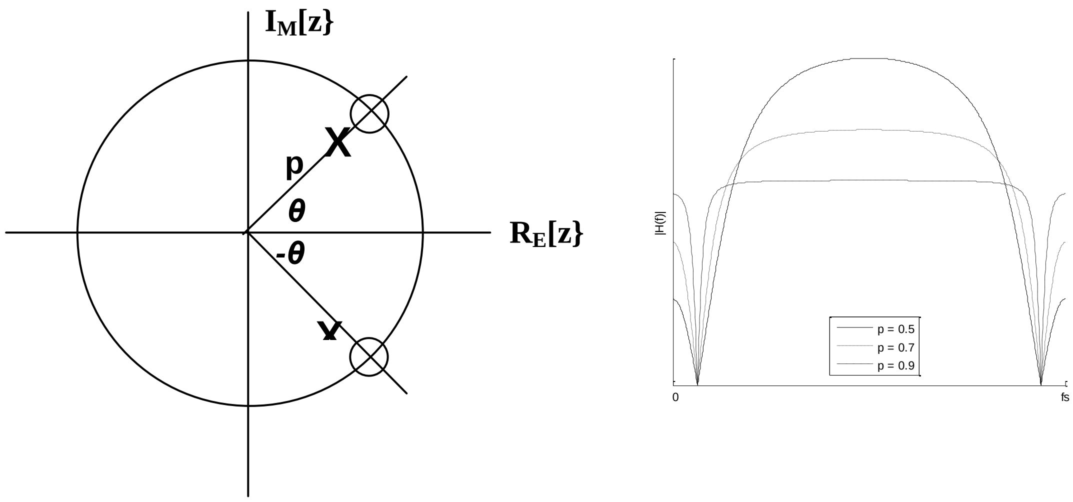


Figure 54. z-plane plot and frequency responses of the simple notch filter for three values of p = 0.5, 0.7 and 0.9.

13.5 Implementing Digital Filters.

It will be clear from much of the foregoing that digital filters are often implemented through recurrence relationships. The outputs are calculated from a group of input samples and a group of previously calculated output samples. When implemented in this way they require only a few lines of computer code and are very fast. There are a number of important points associated with this type of operation.

- The filter must be started off some way into the input data array to avoid addressing outside of the array bounds. For example, our lowpass moving average should be started at $x(N+1)$ so that the $x(n\text{-}N)$ term becomes $x(1)$.

- The output data array should be initialised to zero before the filtering operation starts.

- When using recurrence relationships to filter very long sequences (greater than 10k, say) there is a danger that computational rounding errors will accrue, causing drift and distortion in the output data. This can be avoided by sectioning the data into shorter sequences, by applying the filter using direct convolution in the time domain, or by using the FFT in the frequency domain.

When using a digital filter for the first time it will make sense to test it by computing its impulse and frequency responses. The steps involved are:

- Set up input and output arrays $x(n)$, $y(n)$, say 256 points long and initialise them to contain all zero values.

- Set $x(M) = 1.0$. M is chosen to be sufficiently far into the beginning of the array to take account of terms such as $x(n\text{-}N)$, $y(n\text{-}p)$ etc.

- Run the filter on array $x(n)$ to produce output array $y(n)$. For the moving average we have the following operation, with $N = 4$.

  For $n = 5$ to 256
  $y(n) = y(n-1) + x(n) - x(n-4)$
  End

- Calculate the modulus FFT of y(n) which now contains the required frequency response.

- Plot the result.

14. Equipment Connections.

For sensors and processes dealing with low frequency signals (less than 1 MHz, say) our primary concern is to minimise noise. This is achieved by screening, typically using coaxial cable in which the outer sheath acts as both screen and earth connection. Alternatively or in addition we may use a differential circuit configuration in which the signal appears on a pair of conductors in close proximity and is not directly referred to ground. The twisted pair, still used in many data and telephone circuits is a common example.

When we wish to operate at high frequencies, typically above 1 MHz, there are problems due to the capacitance and inductance of the conductors. As an example, consider the circuit shown on figure 55.

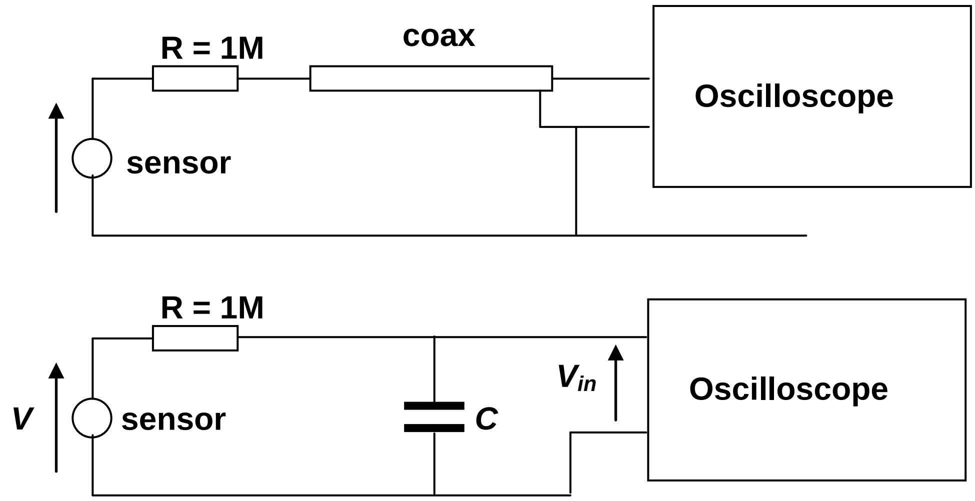


Figure 55. (Top) A sensor connected to an oscilloscope via a coaxial cable and (bottom) the equivalent circuit.

It shows a sensor delivering a signal voltage V 'behind' its output resistance R = 1MΩ; this sensor is connected to an oscilloscope by a coaxial cable of length 1m, and this has a capacitance *C* between its inner and outer conductors of around 100pF. The equivalent circuit is also shown on figure 55. The voltage at the oscilloscope will be a fraction of the sensor voltage given by

$$\frac{V_{in}}{V} = \frac{\omega_0}{s + \omega_0} \qquad where \quad \omega_0 = \frac{1}{CR} \qquad (120)$$

The effective cut off frequency will be

$$f_0 = \frac{1}{2\pi CR} \quad = \quad \frac{1}{2\pi \times 100 \times 10^{-12} \times 10^6} \quad = \quad 1.6\,\text{kHz}.$$

Thus frequencies of interest in the MHz range will effectively be lost. In practice this problem is overcome by using impedances matched to those of the connecting coaxial cables - we consider the latter as transmission lines.

14.1 The Transmission Line.

The coaxial cable with which we connect our equipment together is considered as a transmission line with a certain capacitance per unit length, *C*, and a certain inductance per unit length, *L* , figure 56.

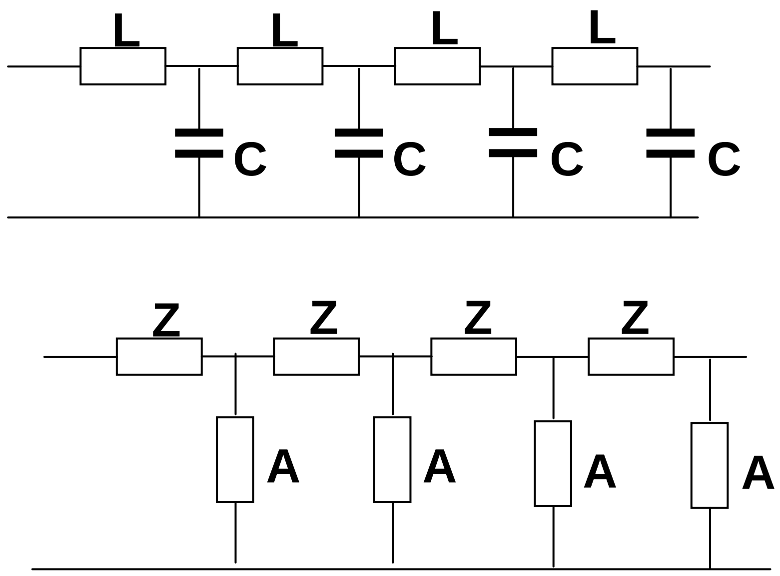


Figure 56. (Top) Equivalent circuit of an electric transmission line, and (bottom) a generalised equivalent ladder network.

To analyse this circuit we begin with the equivalent generalised ladder; the voltage drop per unit length is

$$\frac{\partial v}{\partial x} = -iZ \quad or \quad \frac{\partial^2 v}{\partial x^2} = -\frac{\partial i}{\partial x} Z \,. \tag{121a}$$

The current drop per unit length is

$$\frac{\partial i}{\partial x} = -Av \tag{121b}$$

Combining equations we get

$$\frac{\partial^2 v}{\partial x^2} = AZv \tag{121c}$$

Now in Fourier transform notation

$A = j\omega C \quad and \quad Z = j\omega L$, whence

$$\frac{\partial^2 v}{\partial x^2} = -\omega^2 LCv = (j\omega)(j\omega)LCv \tag{121d}$$

Now each ($j\omega$) corresponds to differentiation in the time domain, whence

$$\frac{\partial^2 v}{\partial x^2} = LC\frac{\partial^2 v}{\partial t^2}\,. \tag{121e}$$

This is the transmission line *wave equation*, and the propagation velocity is given by

$$c^2 = \frac{1}{LC} \tag{121f}$$

In most coaxial cables c is approximately 2×10^8 ms^{-1}. Electric signals on our line behave just as the ultrasonic signals which have been covered earlier in the Bristol University contribution to this course. Importantly they are subject to reflection at discontinuities in the electric impedance of the line. To analyse reflection phenomena we need the *characteristic impedance* of the line, derived as follows:

The voltage wave is

$$v = e^{j\omega t} e^{-jkx} \tag{122a}$$

Where k is the wavenumber given by

$$k = \frac{\omega}{c} = \omega\sqrt{LC} \tag{122b}$$

The current on the line is

$$i = -\frac{1}{Z}\frac{\partial v}{\partial x} = \frac{-1}{j\omega L} e^{j\omega t}(-jk)e^{-jkx} \tag{122c}$$

The characteristic impedance of the line is

$$Z_L = \frac{v}{i} = \frac{j\omega L}{jk} = \frac{j\omega Lc}{j\omega} = \frac{L}{\sqrt{LC}} = \sqrt{\frac{L}{C}} \tag{122d}$$

If the line is terminated by an impedance Z_T part of the signal is reflected back up the line, the reflection coefficient being

$$r = \frac{Z_T - Z_L}{Z_T + Z_L} \tag{122e}$$

The reflected wave will travel back up the line to the source where, if the source impedance is not the same as the line, it will reflect again. The result will be a series of reverberations on the line which enter the load – probably an oscilloscope or other recording device. The signal actually recorded will be a distorted, that is, filtered, version of the original signal.

In order to avoid these distortions it is imperative that the device at which the transmission line is terminated has an impedance equal to that of the line – that is to say, it is *matched* to the line impedance. Most coaxial cables are of 50 Ohm impedance, so inputs to recording devices should have an input impedance close to 50 Ohm as well. If they do not, and, typically exhibit an input impedance of 1 M Ohm then a BNC terminator pod should be used between the line and the recorder. Terminators contain π-configuration circuits such as is illustrated in figure 57.

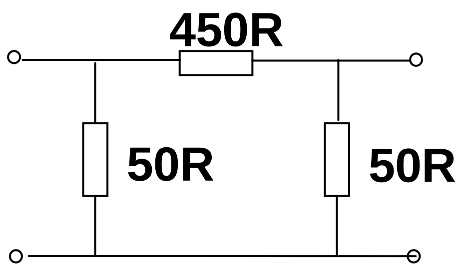


Figure 57. Basic circuit of a line terminator; the component values are approximate.

The coaxial cable is connected to the input terminal where it will ‘see’ a load of 50 Ohms, with a much greater resistance in parallel. The signal to the output is attenuated by approximately factor 10 and there will thus be a compromise between the need to match connections and some signal attenuation, in this case 20dB. Line terminators can be obtained with various levels of attenuation.