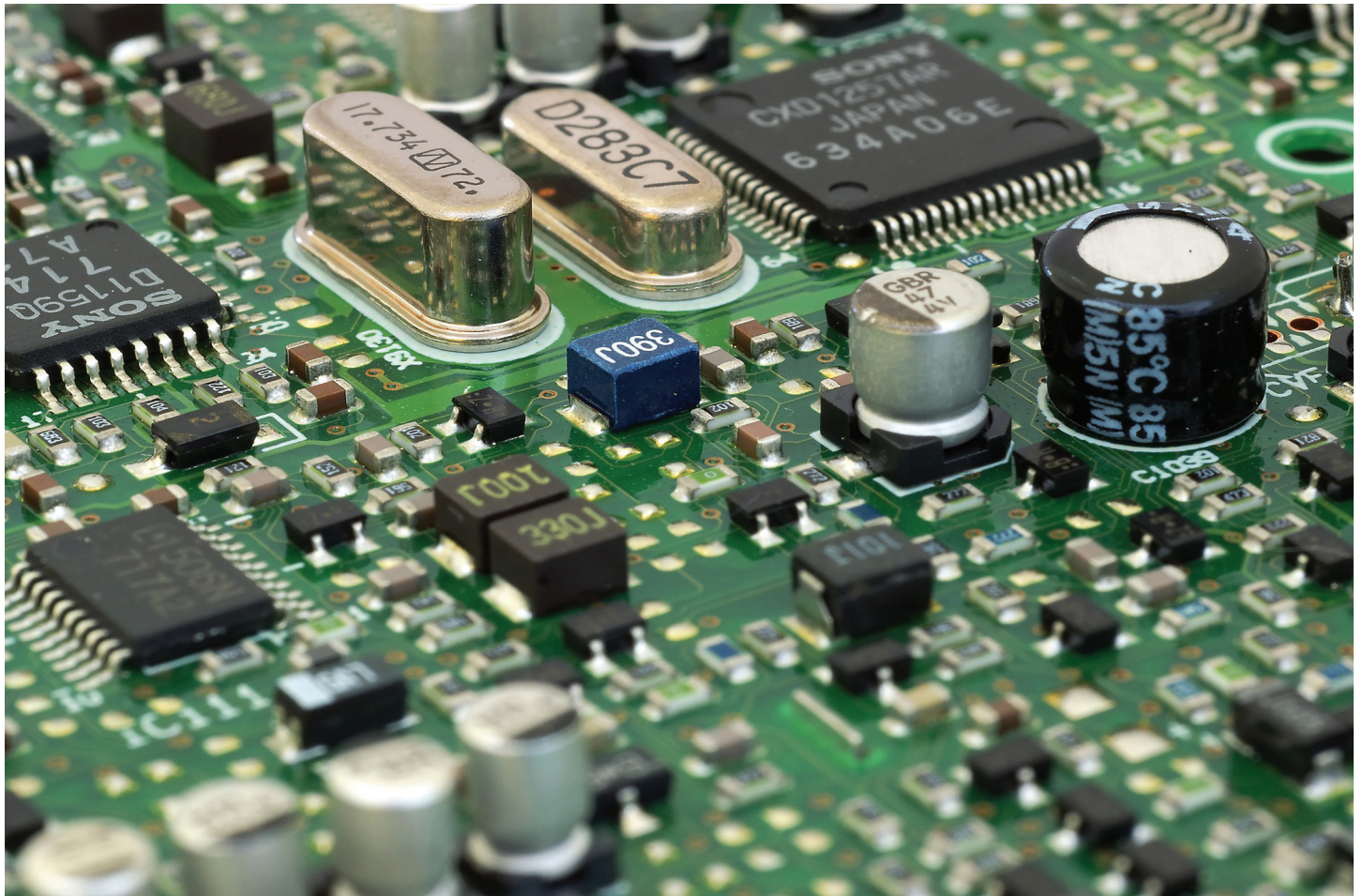


Electronic applications



This item contains selected online content. It is for use alongside, not as a replacement for the module website, which is the primary study format and contains activities and resources that cannot be replicated in the printed versions.

About this free course

This free course is an adapted extract from the Open University course T312 *Electronics: signal processing, control and communications* - <http://www.open.ac.uk/courses/modules/t312>.

This version of the content may include video, images and interactive content that may not be optimised for your device.

You can experience this free course as it was originally designed on OpenLearn, the home of free learning from The Open University –

[Electronic applications](#)

There you'll also be able to track your progress via your activity record, which you can use to demonstrate your learning.

Copyright © 2020 The Open University

Intellectual property

Unless otherwise stated, this resource is released under the terms of the Creative Commons Licence v4.0 http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en_GB. Within that The Open University interprets this licence in the following way:

www.open.edu/openlearn/about-openlearn/frequently-asked-questions-on-openlearn. Copyright and rights falling outside the terms of the Creative Commons Licence are retained or controlled by The Open University. Please read the full text before using any of the content.

We believe the primary barrier to accessing high-quality educational experiences is cost, which is why we aim to publish as much free content as possible under an open licence. If it proves difficult to release content under our preferred Creative Commons licence (e.g. because we can't afford or gain the clearances or find suitable alternatives), we will still release the materials for free under a personal end-user licence.

This is because the learning experience will always be the same high quality offering and that should always be seen as positive – even if at times the licensing is different to Creative Commons.

When using the content you must attribute us (The Open University) (the OU) and any identified author in accordance with the terms of the Creative Commons Licence.

The Acknowledgements section is used to list, amongst other things, third party (Proprietary), licensed content which is not subject to Creative Commons licensing. Proprietary content must be used (retained) intact and in context to the content at all times.

The Acknowledgements section is also used to bring to your attention any other Special Restrictions which may apply to the content. For example there may be times when the Creative Commons Non-Commercial Sharealike licence does not apply to any of the content even if owned by us (The Open University). In these instances, unless stated otherwise, the content may be used for personal and non-commercial use.

We have also identified as Proprietary other material included in the content which is not subject to Creative Commons Licence. These are OU logos, trading names and may extend to certain photographic and video images and sound recordings and any other material as may be brought to your attention.

Unauthorised use of any of the content may constitute a breach of the terms and conditions and/or intellectual property laws.

We reserve the right to alter, amend or bring to an end any terms and conditions provided here without notice.

All rights falling outside the terms of the Creative Commons licence are retained or controlled by The Open University.

Head of Intellectual Property, The Open University

Contents

Introduction	4
Learning Outcomes	5
1 Electronics everywhere	6
1.1 Three key subsystems	7
2 Signal processing	10
2.1 Frequency-dependent gain	10
2.2 Gain functions of ideal filters	11
2.3 Types of interference	12
2.4 First-order filters	14
2.5 Normalised first-order low-pass filters	15
2.6 The full Bode plot: gain and phase	18
2.7 Chebyshev and Butterworth filters	19
3 Digital signal processing	21
3.1 Digital filters	22
3.2 Characteristics of discrete-time and continuous-time signals	23
3.3 Sampling a continuous-time signal	24
3.4 Quantisation of a signal	25
3.5 Digital filtering in the time domain	27
3.6 Designing a digital filter in the frequency domain	29
3.7 Fourier transforms and the sinc pulse	31
3.8 A low-pass filter design	33
3.9 Digital filtering in practice	38
3.10 Adding noise	39
3.11 Changing the decision level	40
3.12 Applying the FIR filter	42
Conclusion	44
Glossary	44
References	46
Acknowledgements	47

Introduction

The modern world would not be able to function without electronic systems. Using a variety of teaching material, including videos and interactive activities, this free course, *Electronic applications*, will show you how electronic systems can be found everywhere in communications, control and signal processing. It focusses on electronic filters, particularly digital filters.

Note that the interactive activities have been designed to work in the Firefox and Chrome browsers, so you will need to use one of these browsers if you want to access the interactive content.

This OpenLearn course is an adapted extract from the Open University course [T312 Electronics: signal processing, control and communications](#)

Learning Outcomes

After studying this course, you should be able to:

- understand the mathematical representations and techniques for manipulating of signals in the time and frequency domains
- explain the application, benefits and limitations of communications, control and signal processing techniques in real world applications
- select and apply appropriate techniques to the analysis of time-varying signals represented in both the time and frequency domain
- use a digital filter to remove Gaussian noise from a signal.

1 Electronics everywhere

Begin by considering the following situation:

You are sitting at home watching the television, and a woman in a diving suit, surrounded by sharks is speaking directly to you, live on television from the Caribbean (Figure 1).



Figure 1 Swimming with sharks

It is possible to take this type of situation for granted, but when thinking about it more closely you might wonder how this is possible. How can someone swimming underwater in the Caribbean be seen and heard by you in your house in the UK?

In this course you will start thinking about everything that's involved in these electronic systems – from signal processing to control and communications – which allow you to watch a live broadcast from the Caribbean.

If you have studied some electronics before, you may be familiar with analogue components such as resistors, capacitors and inductors, as well as more complex components such as the operational amplifier. You may also know something about digital circuits, which include logic gates, microprocessors and even software. And you may be aware of some of the fundamental principles that describe the way that voltages and currents in a circuit are related, including Ohm's law and Kirchhoff's voltage and current laws.

Whilst all of this is necessary to understand electronics, it doesn't really explain how electronics allows you to watch a live broadcast from the Caribbean. In part, this is due to the sheer scale of electronic systems. A typical computer in 2020 has around 10 000 000 000 (or 10 billion) transistors. Nobody could sit down and design such a computer armed with Ohm's and Kirchhoff's laws alone. Electronic systems have to be broken down into subsystems, and these subsystems in turn have to be broken down into further subsystems. Go far enough and you'll find that you can explain what is going on using Ohm's and Kirchhoff's laws, but that's a long way down.

At a higher level, when designing subsystems, descriptions like 'the control subsystem' or 'the communications subsystem' are used. These are designed using principles that

describe the function of the subsystem. An electronics engineer would use these system-level principles to design the subsystem, then implement it electronically.

So, what are the systems and subsystems that allow you to watch someone swimming with sharks live on television?

1.1 Three key subsystems

In order to watch someone swimming with sharks live on television somebody or something must be operating the camera and recording the sounds. The images and sounds are converted into electronic signals; these signals are then sent as electromagnetic waves, via satellite, to your satellite dish at home. This then delivers the electronic signals to your television, which converts them back into images and sounds. As you can see, there are several subsystems involved:

- sensors – converting images and sounds into electronic signals
- processors – converting the electronic signals into a form that can be stored and transmitted as radio waves
- communication – actually transmitting the waves, and receiving them at the other end
- processors – converting the waves back into images and sounds
- display – your television.

These components can be broadly categorised into signal processing and communications.

- **Signal processing** explains how signals are manipulated electronically to filter out noise and to alter the signals so that they can be communicated.
- **Communications** looks at how the communication subsystems work, showing how the electronic signals are converted to radio waves for transmission and reception.

Are there any other possible sub-systems?

Yes, there are many, but the one other major area of electronics is **control**. In the scenario above, it's not obvious how control comes into it, however there are probably control systems involved, carrying out tasks such as the auto-focusing function on the cameras.

Another and perhaps more obvious way you could introduce control would be to make your viewing experience more interactive. What if you could control the camera remotely so that you decide what you want to see? Is that so far-fetched? There are already systems which allow you to control your home from your phone – you can adjust the heating, switch on lights, and even see who is at the door. So why not have the full interactive experience of seeing and hearing what you want to see and hear by controlling the live camera equipment through your television?

What has been described here so far is mainly for entertainment. However, the same principles can be applied to other systems, such as a Mars rover (Figure 2). This is a vehicle that travels around the surface of Mars semi-autonomously, collecting samples and analysing them, then sending the data back to Earth. Such a system is clearly making use of signal processing and communication, but it is also using control subsystems to move the vehicle around.

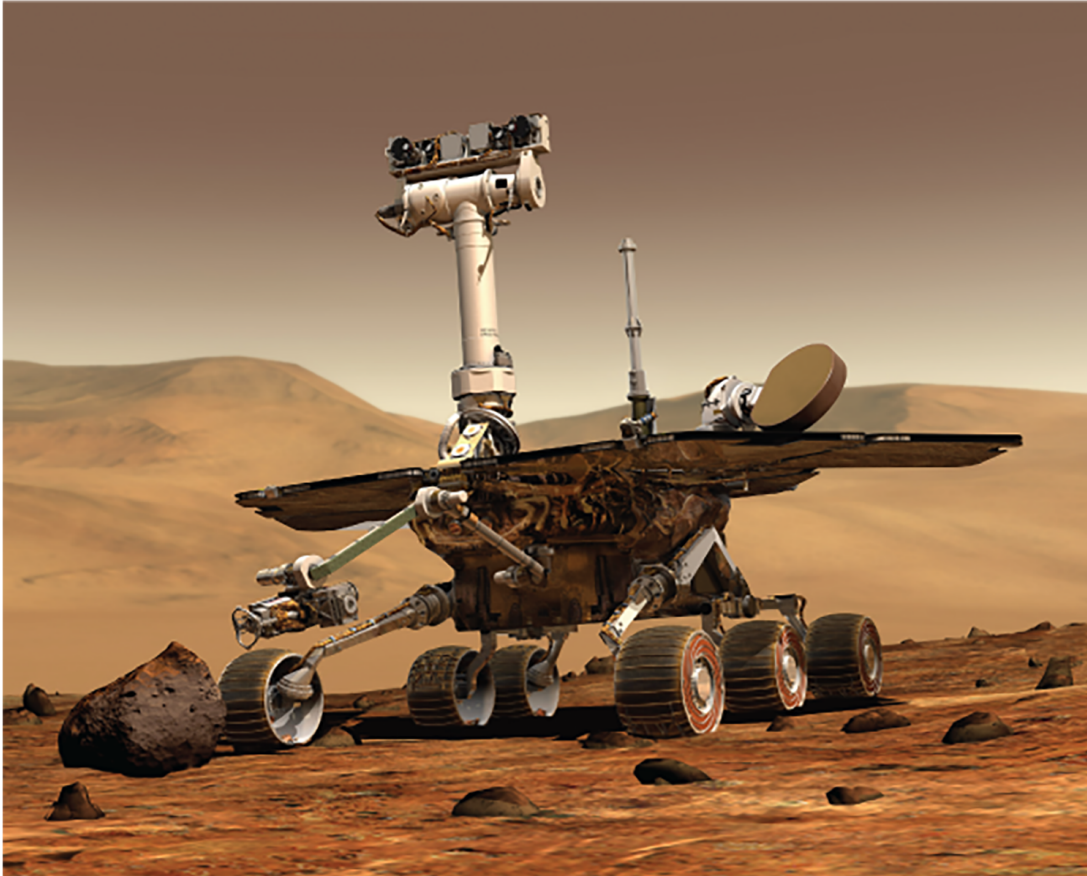


Figure 2 Mars Rover

Systems that use signal processing, control and communications are not just found on other planets. On Earth there are similar systems, including semi-autonomous delivery robots that can bring goods to homes or places of work (Figure 3).



Figure 3 Delivery robot

For the rest of this course you will focus on one of these three subsystems: signal processing. You will look at some of the basic principles of signal processing and how it is implemented.

2 Signal processing

Signal processing is a branch of electronics concerned not just with the properties of signals, but also with the properties of the devices and systems that carry the signals. The objective of signal processing is to optimise the recovery of some particular aspect of the signal that is of most interest, or to optimise the use of a communication medium.

Signal processing usually means filtering a signal. This could be to reduce or remove interference; it might mean changing the signal so that the communication channel can be used more economically or more efficiently; or it might mean processing a signal so that it can be sampled satisfactorily in an analogue-to-digital converter.

What is filtering?

In the context of electronic signals, filtering means altering the signal so that some aspects of the signal are removed while other parts of the signal remain. In this section you will learn about the difference between ‘ideal’ filters and real filters. You will also look at different types of filters and their characteristics and a type of graph used to show the frequency-dependent gain of a filter.

One of the most common uses of filters is to reduce the intrusion of *unwanted* signals, or noise, into *wanted* signals. A common example of filtering is radio and television tuning. Here, the antenna picks up multiple transmissions being broadcast on different frequencies; the tuning circuit ideally passes the wanted broadcast to the output and blocks all the others, which are on different frequencies from the wanted broadcast.

Filters can be analogue or digital. Analogue filters use components such as inductors, capacitors, resistors, and sometimes **operational amplifiers (op-amps)**. Digital filters are basically computers and achieve their filtering effects through mathematical operations on the sample values of a digitised signal. You will look at these more closely in Section 3. In this section, however, you will focus only on analogue filters.

2.1 Frequency-dependent gain

In Figure 4, the box in the middle represents a device with frequency-dependent gain – in other words, a filter. Here the symbol is being used to represent angular frequency, measured in radians per second.



Figure 4 Filter with input amplitude and output amplitude

A sinusoidal input signal with amplitude is applied to the filter, and the output is a sinusoidal signal with amplitude . The **voltage gain** of the filter at the frequency in

question is – in other words, it is the ratio of the output voltage amplitude to the input voltage amplitude. So, at any particular frequency,

The voltage gain can be expressed simply as a number or fraction (or decimal). For example, a gain of 2 means that the output amplitude is twice the input amplitude. A gain of (or 0.25) means that the amplitude of the output is one-quarter that of the input.

The above gain is referred to as ‘voltage gain’ because gain is sometimes expressed as a ratio of output and input *powers*. In this course this way of expressing gain will be referred to as **power gain**. As power ratios can be expressed in decibels, so power gains are almost invariably given in **decibels**.

For the time being you will only consider sinusoidal inputs and outputs, as these have a single frequency. This limitation to a single frequency helps to clarify what a filter does. In practice, though, a filter would typically operate on a complex waveform consisting of many frequency components. In such a case, the inputs and outputs would themselves be functions of frequency.

Although two different types of gain have been mentioned (voltage gain and power gain), you will often just see the word **gain** used by itself. If the gain is given as an ordinary numerical value (such as 2, 10, 3000 or 0.001), voltage gain is almost invariably indicated. If the numerical value is in decibels, power gain is being represented.

The output of a filter differs from the input not only in amplitude but (usually) also in phase. You will look more closely at the question of phase later in the course. However, for now you will continue to focus on gain. Complete Activity 1 to test your understanding so far.

Activity 1

Allow about 5 minutes

- a. A passive filter has an input signal of volts. The steady-state output is volts. What is the gain as a voltage ratio?
- b. The input to the filter in part (a) remains unchanged in amplitude, but its frequency changes. The steady-state output is now found to be volts. What is the new gain as a voltage ratio?

Answer

- a. Here , the amplitude of , is 10 V and , the amplitude of , is 2 V. Therefore the gain is , or 0.2.
- b. is still 10 V and is now 5 V. Therefore the gain is now , or 0.5.

In the next section you will discover the characteristics of an ideal filter.

2.2 Gain functions of ideal filters

Figure 5 shows some common types of ‘ideal’ filter. Ideal filters are sometimes characterised as ‘brick-wall’ filters because graphs of their gain functions have perfectly

horizontal or vertical lines. In practice, such brick-wall gain functions can never be achieved.

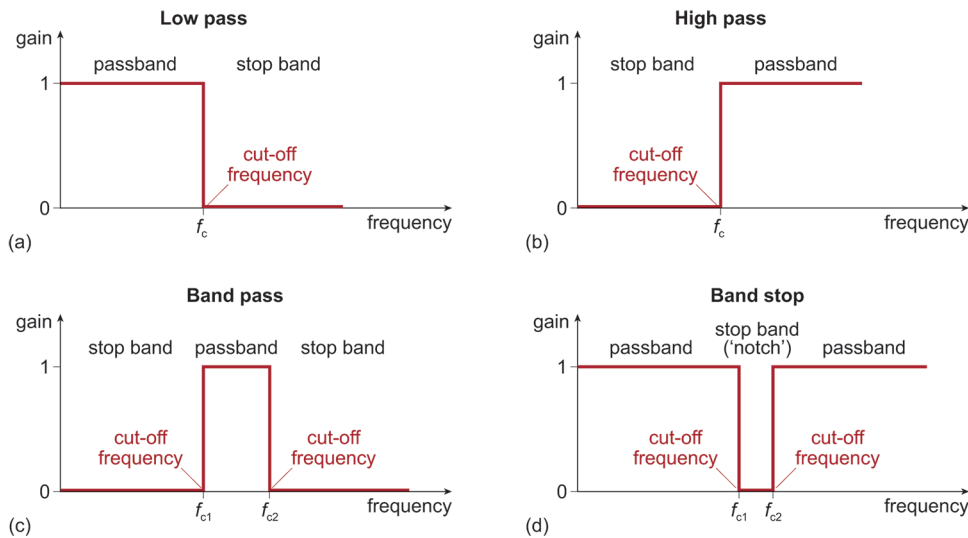


Figure 5 Four types of ideal filter: (a) low-pass filter; (b) high-pass filter; (c) band-pass filter; (d) band-stop (notch) filter

Notice that each of the four types of filter has a name summarising what it does. For example, the low-pass filter (Figure 5(a)) passes all frequencies below the cut-off frequency and blocks all frequencies above it.

In all the filters, a frequency band where the signal is passed is called a **passband**, and a frequency band where the signal is blocked is called a **stop band**. All the filters in Figure 5 have one or more passbands, one or more stop bands, and one or more cut-off frequencies.

In all the passbands shown, the voltage gain is 1, but this is a convention for this type of diagram. The actual passband gain depends on various factors such as whether the filter is passive (that is, consists only of passive components, such as resistors, capacitors and inductors) or active (that is, includes amplification as well as passive components).

Similarly, all the stop bands are shown with a voltage gain of 0. In practice, the gain is likely to be above 0. However, provided the stop-band gain is several orders of magnitude below the passband gain, the term 'stop band' is reasonable.

In the next section you will see what is meant by interference and noise, which is what is trying to be removed from a signal using a filter.

2.3 Types of interference

In the same way that Figure 5 shows simplified models of filters, there exist simplified models of the type of signals and noise we might want to apply filters to.

A common type of interference is adjacent channel interference, in which the interfering signal is in a frequency band above or below that of the wanted signal, as shown in Figure 6.

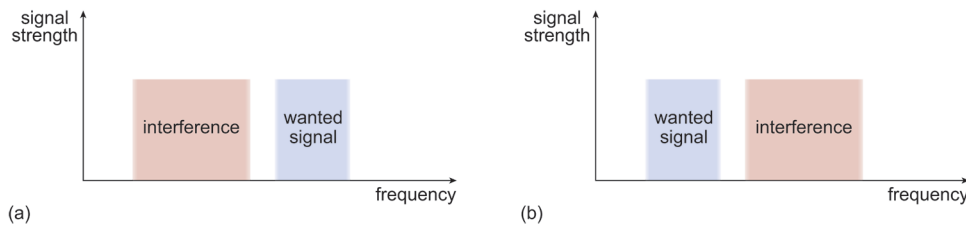


Figure 6 Adjacent channel interference: (a) below wanted signal; (b) above wanted signal

In each case, an appropriate filter can be used to reduce the interference. For example, you can see that in the case of Figure 6(b), where the adjacent channel interference is above the wanted signal, a low-pass filter can be used, with the passband coinciding with the wanted signal and the stop band coinciding with the interference. With this arrangement an ideal filter could, in principle, remove the interference altogether. (However, as you will see later, the reality is somewhat different.)

Life is trickier when the signal and interference overlap in frequency, as in the narrowband interference and wideband interference shown in Figure 7. Here, no form of filtering can give us what we would like, which is a noise-free signal with no adverse effect on the signal.

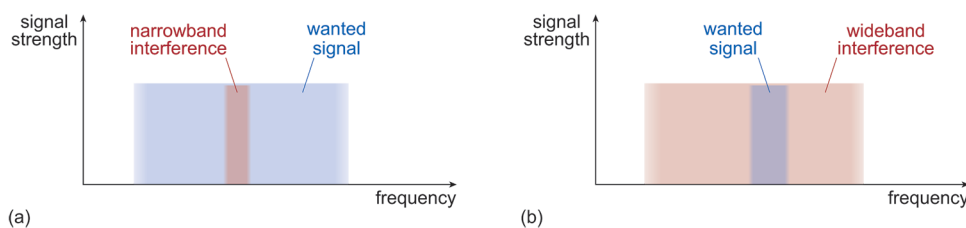


Figure 7 (a) Narrowband interference; (b) wideband interference

Note that, just as brick-wall filters are unachievable in practice, the brick-wall frequency bands and signal strengths of these various types of interference are not achievable in practice. In real-world situations, the boundaries are less clearly defined.

Now complete Activity 2 and apply the correct filter to the interference type.

Activity 2

Allow about 5 minutes

For each of the following types of interference, suggest a suitable filter to improve the signal-to-noise ratio, and say how the passbands and stop bands should be arranged. Explain any drawbacks.

- Narrowband interference
- Wideband interference

Answer

- For narrowband interference you can use a band-stop filter, with the stop band centred on the interference and of an equal width. This could in principle remove the interference. The drawback is that the stop band also removes some signal power.
- When you have wideband interference, a suitable remedy is to use a band-pass filter with the passband centred on the wanted signal and equal in width to the

bandwidth of the signal. However, although this gives the best signal-to-noise ratio, it cannot remove all the interference.

Having seen the characteristics of ideal filters and the sorts of interference that you want to remove, the next section will look you what real filters are like.

2.4 First-order filters

In addition to the filter categories already introduced (low-pass, band-pass, etc.), filters are categorised by their **order**. The order of a filter is determined by the form of the **differential equation** governing the filter's behaviour. The simplest type of filter, with the simplest equation, is called a **first-order** filter. **Higher-order** filters are more complex than first-order filters, both in their circuitry and in the differential equation that governs them. The higher the order, the more effective the filter.

An example of a first-order filter is the simple circuit in Figure 8.

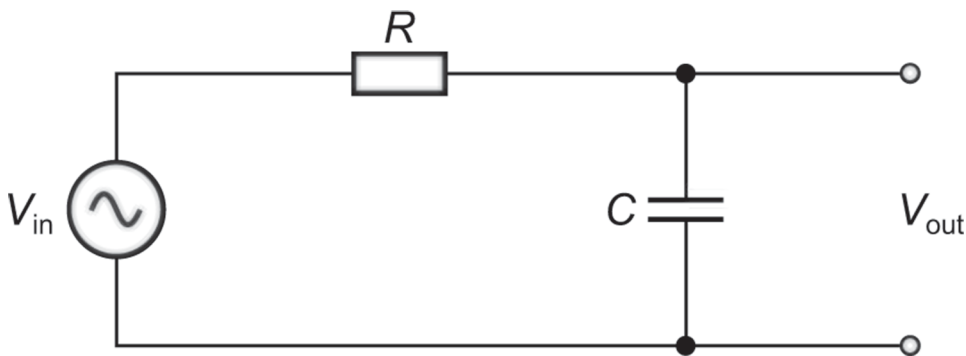


Figure 8 First-order filter

Activity 3

Allow about 5 minutes

Which of the four categories of filter shown in Figure 5 does the filter in Figure 8 belong to? You should be able to work it out from the behaviour of the capacitor at low frequencies and high frequencies. Explain your answer.

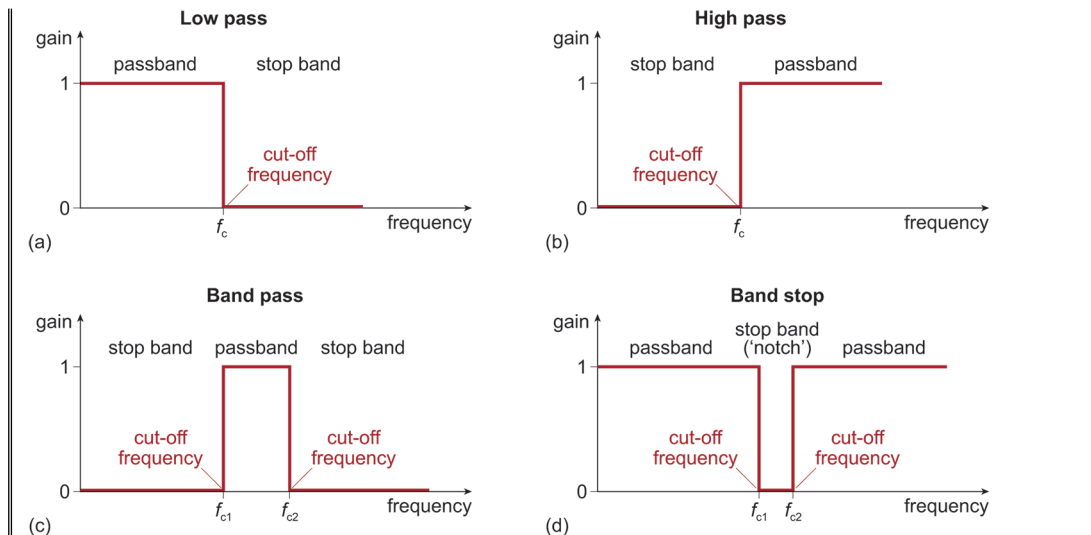


Figure 5 (repeated) Four types of ideal filter: (a) low-pass filter; (b) high-pass filter; (c) band-pass filter; (d) band-stop (notch) filter

Answer

It is a low-pass filter.

At 0 Hz, or DC, the capacitor is open circuit. Under those circumstances, all the input voltage would appear on the output. At high frequencies, the capacitor becomes increasingly like a short circuit, and the output voltage decreases as frequency increases. Therefore low frequencies are passed and high frequencies are (to some extent) blocked.

Low-pass and high-pass filters can be first-order, second-order, third-order, and so on. However, band-pass filters and band-stop filters must be second-order or higher, although it is possible to achieve their effect by combining two first-order filters.

Despite being the simplest type, a first-order filter is not simple to analyse mathematically, as its behaviour is governed by a first-order differential equation. This is what the voltage gain function of the filter in Figure 8 turns out to be:

You could plot a graph of the gain function against frequency from this equation, but the details would depend on the choice of ω_c and K . However, irrespective of the values chosen for ω_c and K , the general shape and certain other properties of the gain function would be the same for any first-order low-pass filter. Therefore, rather than show how the gain function changes with frequency for particular values of ω_c and K , in the next section you will look at a 'normalised' version of the gain function. (Normalisation is the presentation of information in a generalised way that can easily be adapted to specific cases.)

2.5 Normalised first-order low-pass filters

A graph of gain function against frequency is called a **frequency response** or a **Bode plot**. A normalised first-order low-pass frequency response (or Bode plot) is shown in Figure 9.

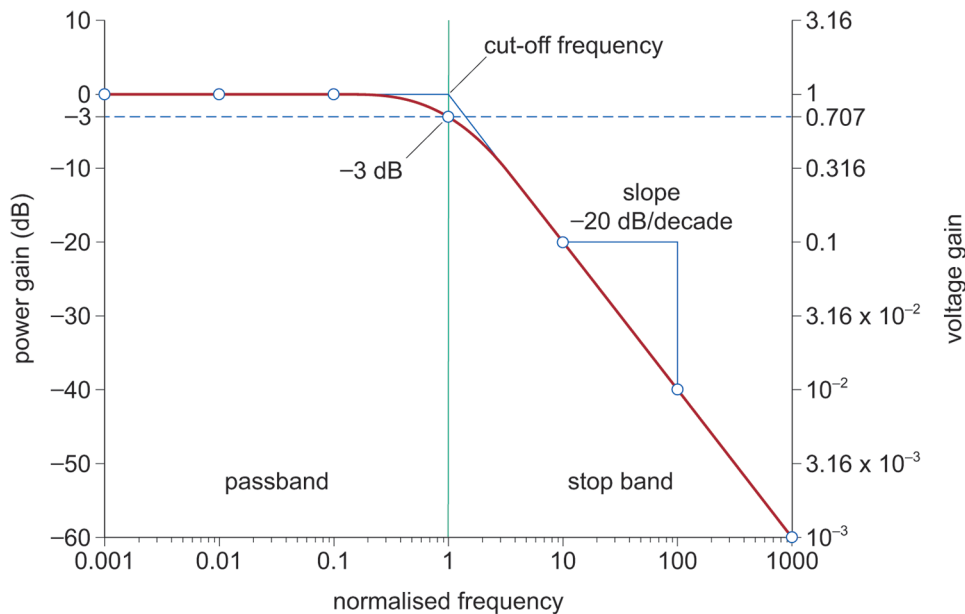


Figure 9 Normalised first-order low-pass frequency response

You can see in the graph that on the horizontal axis, the frequency scale is in a form that is known as logarithmic. What this means is that at equally spaced points instead of having 1, 2, 3 Hz etc., you have 1, 10, 100 Hz etc. The frequency increases by a factor of 10 at each interval.

Secondly, the vertical axis shows the power gain, but is measured in decibels. This again is a logarithmic measure which is designed to measure the ratio of powers, but is probably more commonly known in the measurement of sound.

The reason logarithmic axes are used is so that the shape of the graph is very nearly made up of a horizontal straight line in the pass band, and a sloping straight line in the stop band.

Just from the shape of the graph, it is clear that this gain function of a real filter is very different from the idealised ones that you saw earlier. For example, there is no sharp distinction between the passband and the stop band, and consequently no distinct cut-off frequency. By convention, the frequency at which the power gain drops 3 dB below the passband gain (or, in terms of voltage ratio, falls to 0.707 times the passband gain) is called the cut-off frequency, and this is true also for an active filter where the passband gain is likely to be other than 1.

In this normalised graph, the frequency axis is the normalised part. Notice that the frequency axis has no units, and the numbers on it are relatively low. (In electronics, such apparently low frequencies are seldom of interest; instead typically frequencies ranging from hundreds of hertz to gigahertz are more often used.) The normalised frequencies have been calculated by dividing actual frequencies by the cut-off frequency. This is why the cut-off point sits at 1 on the normalised frequency axis.

Dividing one frequency by another results in a pure number (that is, one without units), which is why the normalised frequency axis has no units. Similarly, dividing an output voltage by an input voltage produces a gain figure that also has no units. If the gain is converted to a decibel value, then strictly speaking that too has no units, although in Figure 9 'dB' has been added after the number as though it were a unit as a reminder of the logarithmic nature of the function used.

As an example of how to translate normalised frequencies to actual frequencies, suppose a practical low-pass filter had a cut-off frequency of 10^4 radians per second. Simply multiplying all the numbers on the horizontal axis of the normalised graph by 10^4 and giving the unit as 'radians per second' would transform the graph into one showing gain against actual frequency for that particular filter. If the cut-off frequency were 10^4 Hz (rather than radians per second), the procedure would be just the same: multiply all the numbers on the axis by 10^4 and give the unit as Hz.

All first-order low-pass filters have a gain function of this shape and with these slopes. The passband gain might differ if the filter is active (that is, if it incorporates amplification). For first-order passive filters (that is, those without amplification) the passband gain cannot exceed 1.

Activity 4

Allow about 10 minutes

An alternating voltage source is connected to the input of a low-pass filter. What power is drawn from the output of the filter if the voltage source operates:

- well below the cut-off frequency of the filter and supplies 0.2 W to the input of the filter?
- at the cut-off frequency of the filter and supplies 0.1 W to the input of the filter?

Answer

- Well below the cut-off frequency, the gain of the filter is 0 dB, which is a power ratio of 1:1. Hence the power drawn from the output of the filter is the same as that supplied at the input, 0.2 W.
- At the cut-off frequency, the power gain is -3 dB or a half. Hence the output power is half the input power, or 0.05 W. The difference between the input power and the output power is also 0.05 W, which is dissipated as heat.

Notice that as you move right along the frequency axis, the gain never reaches zero (that is, zero as a voltage ratio rather than as a decibel number). Zero gain would correspond to a negatively infinite number of decibels.

All real first-order filters (as opposed to ideal ones) lack a sharp cut-off frequency; in addition, low-pass filters never fully cut off, although if the gain is low enough you can regard it as having fallen to 0. Higher-order filters can give a sharper cut-off than a first-order filter, but the brick-wall cut-off of an ideal filter can never be achieved in practice.

As you move left along the frequency axis, each division is one-tenth of the one before, and this can continue indefinitely. A logarithmic frequency scale therefore never reaches a frequency of 0.

The steepness of the gain in the stop band is referred to as the filter's **roll-off**. All first-order filters have a 20 dB/decade roll-off. The same roll-off can also be specified as 6 dB/octave. An **octave** is a term borrowed from music and represents a doubling of frequency. (It is so called because the frequency span in a doubling of frequency is divided into the eight notes of a musical scale.) Higher-order filters have a steeper roll-off. For second-order filters it is 40 dB/decade (or 12 dB/octave) and for third-order filters it is 60 dB/decade.

decade (or 18 dB/octave). Each successive order adds a further 20 dB/decade (or 6 dB/octave) to the roll-off.

Note that at the first decade above the cut-off frequency, the gain of a first-order filter is 20 dB below the passband gain, not 20 dB below the gain at the cut-off frequency.

Now have a go at Activity 5.

Activity 5

Allow about 10 minutes

A first-order low-pass filter has a cut-off frequency of 6.28×10^3 radians per second. !Warning! inherit not supported³

- What is the cut-off frequency in hertz? (Round your answer to 2 significant figures.)
- What is the gain at 1 kHz? Express your answer in decibels.
- What is the gain at 100 kHz? Express your answer in decibels.

Answer

- Angular frequency, where is the frequency in hertz. So

To 2 significant figures, $\omega = 10^3$ Hz. So the cut-off frequency is 10^3 Hz or 1 kHz.

- 1 kHz is the cut-off frequency, so the gain here is -3 dB.
- 100 kHz is two decades above the cut-off frequency, so the power gain is -40 dB.

As you saw earlier, filters affect the gain and phase of a signal. In the next section you will see how you can show this in the Bode plot.

2.6 The full Bode plot: gain and phase

In Section 2.5, you heard how a full Bode plot would show not only how the gain changes with frequency, but also how the phase difference between output and input changes with frequency. Conventionally the phase plot is put under the gain plot, with their respective frequency axes aligned, as in Figure 10. This example is for a first-order low-pass filter.

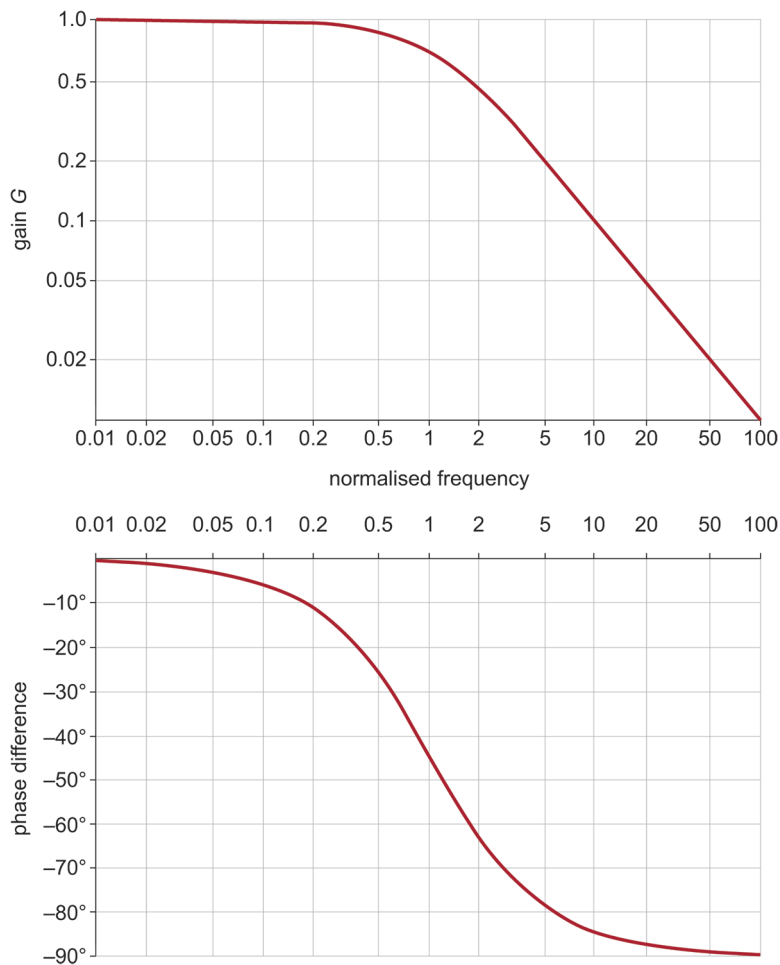


Figure 10 Normalised first-order low-pass frequency response showing gain and phase

Figure 10 shows that in the passband, the output is virtually in phase with the input. As frequency increases towards the cut-off frequency (1 on the normalised frequency axis), a phase difference opens between the output and the input, with the output lagging the input (the negative values of phase angle indicate lagging phase). At the cut-off frequency, the phase difference is -45° . This means that for a sinusoidal input, the output lags the input by 45° . By the time you are well into the stop band, the phase difference levels off at -90° .

That concludes the discussion of first-order filters. To end Section 2 you will consider some commonly found higher-order filters and have a look at their characteristics.

2.7 Chebyshev and Butterworth filters

The design of higher-order filters is a specialist area, and mathematically complex, so in this section you will look at gain functions of just two celebrated types. The first is the Butterworth filter. The gain function of a Butterworth filter has the familiar flat passband and roll-off you would expect. However, this filter comes in various orders, such as 2, 4, 6, 8 and 10, depending on how steep the roll-off needs to be.

Figure 11 shows the normalised responses of first-order, second-order and eighth-order low-pass Butterworth filters. As the order of the filter increases, it approximates more closely the ideal brick-wall response.

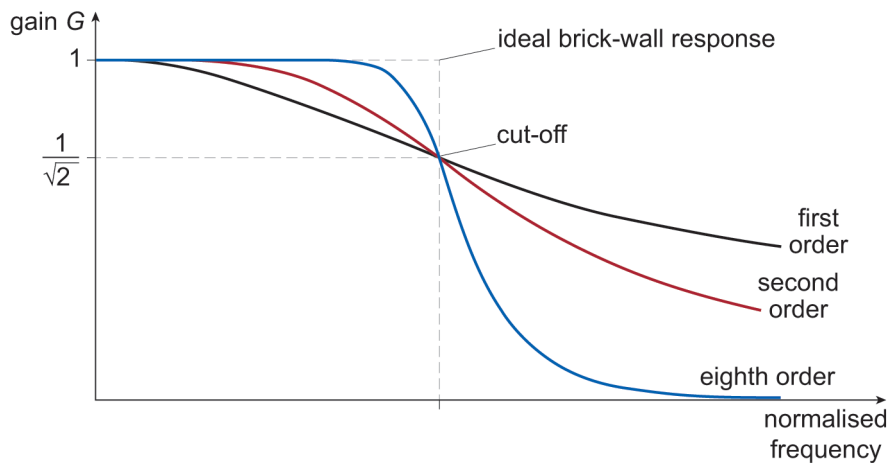


Figure 11 Normalised Butterworth magnitude response curves

Steeper roll-offs than those of a Butterworth filter can be had if other desirable features of a filter are compromised. The Chebyshev filter has a steeper roll-off, but at the price of a passband response that is not flat. Figure 12 compares seventh-order normalised Chebyshev and Butterworth filters. Notice the ripple in the Chebyshev filter's response in the passband. (There is another type of Chebyshev filter that has a flat response in the passband and ripples in the stop band.)

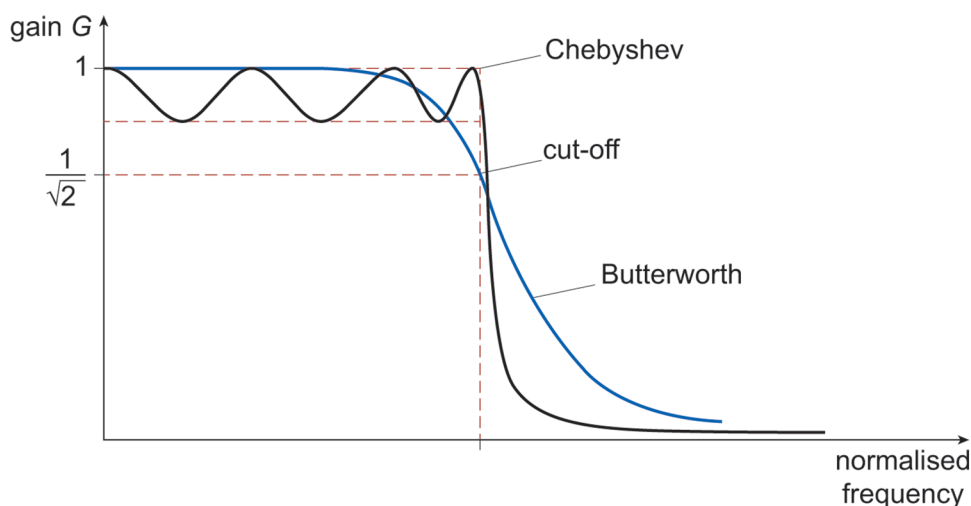


Figure 12 Comparison of seventh-order Butterworth and Chebyshev gain

That concludes the introduction to filters and, in particular, analogue filters. In Section 3 you will learn about digital filtering, which is currently a more popular approach to filtering.

3 Digital signal processing

Digital signal processing has developed rapidly over the last 50 years: digital signal-processing circuits have become faster and cheaper, and memory storage capabilities have increased dramatically. One result of these developments is a migration from analogue to digital circuits for some types of signal processing, and digital filters are an example of this trend.

Digital filters have some advantages over analogue filters. They are programmable, so their operation is determined by a program stored in a processor's memory. This means a digital filter can easily be changed without affecting the hardware. Also, digital filters are extremely stable with respect to both time and temperature. For complex filters, the hardware requirements are relatively simple and compact in comparison to the equivalent analogue circuitry.

The design of a digital filter is complicated and involves quite advanced mathematics, so software tools that produce a filter design from your specification of filter characteristics are commonly used. However, as you may know from the use of software tools such as circuit simulators, you need to be very wary of using a software design tool to create circuits without having a good understanding of the electrical characteristics of the circuit that you want to create and the parameters used in the design process. A good understanding of how digital filters work will help at every stage of filter design.

For the remainder of the course you will find out about various aspects of filtering a signal digitally. First you will see how a continuous-time signal is converted to produce the digital discrete-time signal used as input to the filter. Then you will find out how mathematical operations applied to the discrete-time signal can remove or diminish unwanted aspects of the signal.

To get started, Figure 13 shows the basic set-up of a digital filter. Don't worry too much about the detail at this point – it will be covered later. As part of the filtering process, the continuous input signal must be sampled and digitised using an analogue-to-digital converter (ADC) to produce a sampled discrete signal. The resulting binary numbers, representing successive sampled values of the input signal, are transferred to the processor, which carries out numerical calculations on them. These calculations typically involve multiplying the input values by constants and adding the products together. If necessary, the results of these calculations, which now represent sampled values of the filtered signal, are output through a digital-to-analogue converter (DAC) to convert the signal back to continuous form. Given that the continuous input signal and the filtered continuous output signal are continuous in time, they are often referred to as continuous-time signals. Similarly, the sampled discrete signal and the digitally filtered signal are often referred to as discrete-time signals. (Here the word 'discrete' means 'consisting of separate parts' as opposed to the word 'discreet' which means 'unobtrusive').

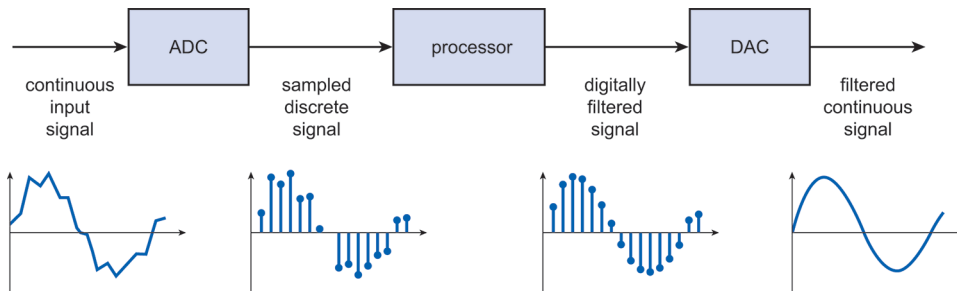


Figure 13 Digital filtering of a sampled signal

In the next section you will see an example of a digital filter being used as part of a medical system to illustrate the component parts.

3.1 Digital filters

You read about analogue filters in Section 2, so you know that they are electronic circuits made up of components such as resistors, capacitors and inductors connected together to produce the required filtering effect. In comparison, a digital filter uses a digital processor to perform numerical calculations on sampled values of the signal to be filtered. The processor could be a general-purpose computer such as a PC, but it is much more likely to be a specialised digital signal processor (DSP) chip, which is designed to carry out the intensive mathematical operations used in digital signal processing quickly and with low power consumption. The low power consumption is important because it means that purpose-designed DSP chips can be used in mobile devices such as phones and tablets.

A potential use for a digital filter with low power consumption is described by Asgar and Mehrnia (2017), who propose using a digital filter in an electrocardiogram (ECG) heart-monitoring system. Figure 14 is a block diagram of the system, taken from their paper. It shows a sensor connected to the surface of the user's skin to monitor their heart function. This signal is then conditioned. Typical actions here might be to amplify and remove aliases from the signal; then an analogue-to-digital (ADC) converter is used to change the signal from analogue to digital form prior to it being filtered.

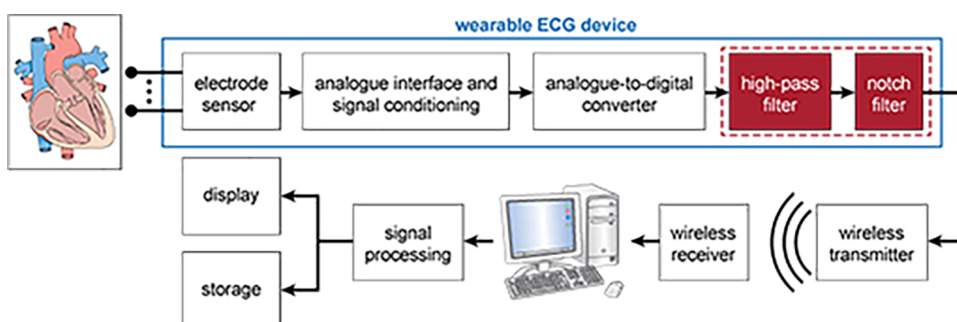


Figure 14 Block diagram of the ECG heart-monitoring system

In the paper, the authors describe six different sources that can cause noise to contaminate the measured signal. Three of these sources are as follows:

- electrode contact noise, which is due to electrode 'popping' or a loose contact with the skin

- instrumentation noise, which is due to radio-frequency interference from other equipment (e.g. implanted devices such as pacemakers)
- electromyographic (EMG) noise, which is induced by electrical activities of skeletal muscles during periods of contraction.

This shows the variety of noise sources that are associated with specific applications and the insights that are needed to understand the noise sources in any filter application. The filtering solution proposed by the authors uses a low-complexity, linear-phase digital filter design.

The next section will describe in more detail the signals that are found in digital systems.

3.2 Characteristics of discrete-time and continuous-time signals

A continuous-time signal is shown in Figure 15(a). The signal is continuous because it has a value at any instance of time – that is, for any value of t , it is possible to read a value of $x(t)$ from the graph. Most signals in the real world are continuous in time. For example, if you were monitoring the temperature of a room, you would be able to take a measured value of temperature at any time.

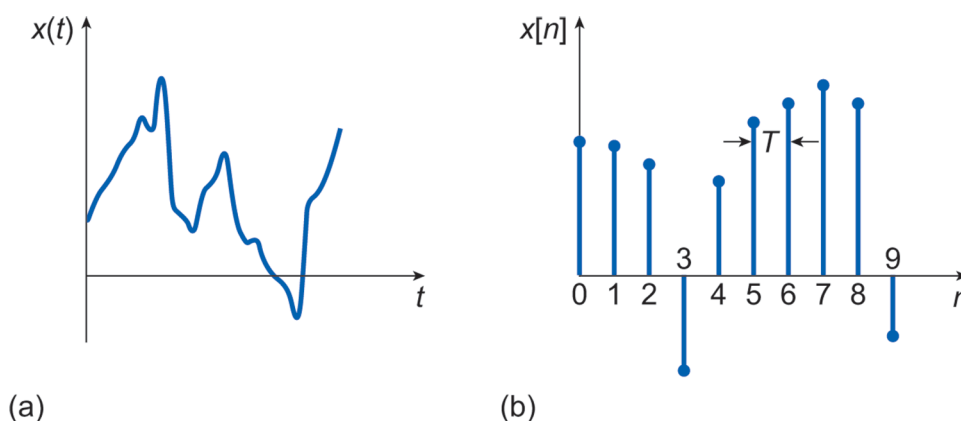


Figure 15 (a) Continuous-time signal; (b) discrete-time signal

A discrete-time signal (sometimes referred to as a time-discrete signal or simply a discrete signal) is shown in Figure 15(b). In the rest of this course the standard convention of drawing the vertical lines in a discrete-time signal with a round dot on the end will be used; these lines-with-dots are often called ‘lollipops’. The signal in Figure 15(b) is discrete because it only has a value at fixed points placed at discrete time intervals seconds apart along the n -axis. This interval is called the sampling interval. Values of $x[n]$ can be found for the integer values of n , such as $n=0, 1, 2, \dots$, but there is no value for the signal at, say, $n=3.5$. Thus n represents the number of the sample.

It is hard to think of examples of real-world discrete-time signals, since most real-world signals are continuous; however, if you took the temperature reading of a room every day at the same time, the result would be a discrete-time signal. Most discrete-time signals come from sampling continuous-time signals to get them into a digitised form that can be processed by digital computers.

Activity 6

Allow about 5 minutes

State whether the following are discrete-time signals or continuous-time signals, giving a reason for each answer:

- the wind speed across the blades of a wind turbine
- the position of a robotic arm as it picks items from a conveyor belt
- the total distance travelled by the robotic arm each hour over a 24-hour period.

Answer

- The wind speed is a continuous-time signal, because you can take a reading at any time.
- The robotic arm always has a position – even if it is in a resting position, you know where it is – so this is a continuous-time signal.
- The total distance travelled by the robotic arm is recorded just once in each hour, so this is a discrete-time signal. Over a 24-hour period there will be 24 discrete values recorded.

In the next section you will learn how a continuous signal is converted to a discrete signal.

3.3 Sampling a continuous-time signal

To convert a continuous-time signal into its discrete-time signal equivalent, you need to sample the waveform. To ensure the discrete-time signal contains the full range of frequencies in the continuous-time signal, the continuous-time signal normally needs to be sampled at a rate that is greater than twice the highest frequency component contained in the signal. The lowest sampling frequency that can fully reconstruct the continuous-time signal is called the Nyquist frequency after Harry Nyquist (1889–1976), a Swedish-born American engineer who made important contributions to communications theory.

What happens if the sample rate is equal to or less than twice the highest frequency in the signal – or, to express this another way, if the continuous-time signal is sampled twice or less than twice within each cycle of the highest frequency component contained in the signal? Figure 16(a) shows a sine wave (solid line) that is being sampled less than twice each cycle. The samples are represented by blobs. However, another sine wave with a lower frequency can be drawn through these samples. It is shown as a dashed sine wave. This lower-frequency wave is called an **alias** of the original sine wave. The way to avoid aliases is to sample more frequently, as in Figure 16(b). Here there are twice as many samples as in Figure 16(a). Now the alias from (a), also shown dashed in (b), misses some of the samples in (b). There is no waveform of a lower frequency than the sampled waveform that can be fitted to all the samples.

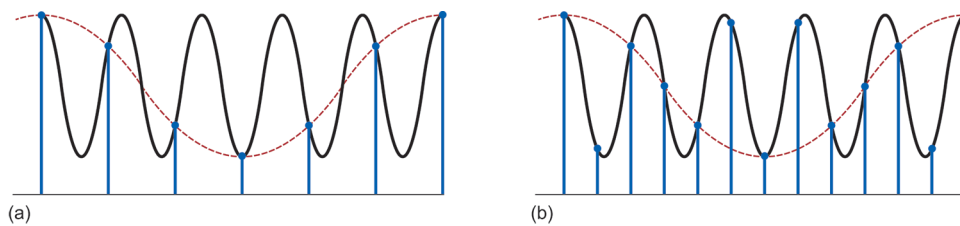


Figure 16 (a) More than one sine wave can be drawn through these samples, so there is a low-frequency alias of the original wave; (b) with more samples, there is no alias that can be fitted to all the samples

The way to eliminate the possibility of an alias is either to have greater than two samples per cycle or, conversely, to ‘band-limit’ the signal to make sure that there are no frequencies equal to or higher than one half of the sampling frequency. This is done by including an analogue filter before the sampler in the analogue-to-digital converter that removes all frequencies higher than a certain frequency. As such, a filter is designed to stop any chance of aliasing occurring, it is sometimes called an **anti-aliasing filter**.

After sampling the signal, the samples need to be converted to a digital signal. The next section describes how this is done.

3.4 Quantisation of a signal

When a continuous-time signal is sampled, the amplitude of each sampled point undergoes quantisation which means that it is forced to have only certain discrete values. The amplitude of each sample is represented by a digital binary code, and the word length of the code will be a fixed number of digital bits. Representing the amplitude of the samples in this way means the value is represented by a binary number, so it is truncated or quantised to its closest binary equivalent.

A 1-bit binary number can represent two levels because it can only take a value of 0 or a value of 1. A 2-bit binary number represents four levels, where each level takes one of the values 00, 01, 10 or 11. Figure 17 shows a discrete-time signal whose values are limited to a 3-bit binary number, which represents eight possible combinations of 0s and 1s – each possible combination is shown on the y -axis of the figure. Note that each sampled, digitised signal is a binary representation of the analogue value, so in conversion to these binary representations some rounding of the signal values has occurred: some sampled values are just above and some just below the continuous-time signal.

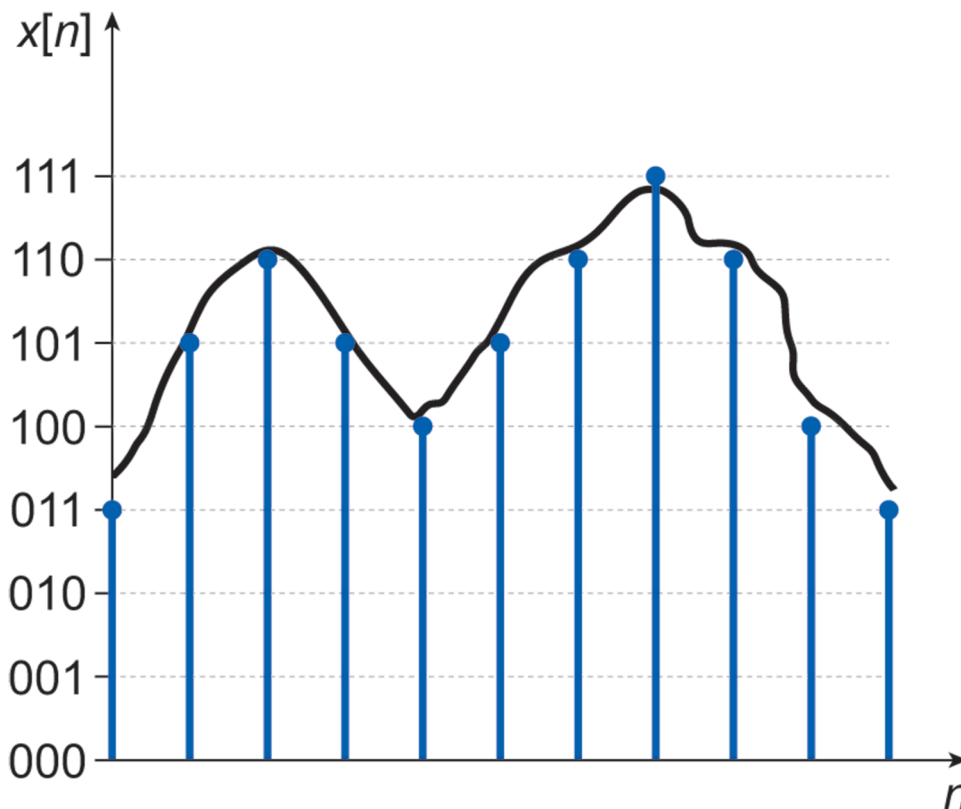


Figure 17 Sampled and quantised signal

Any binary word will always give an even number of **quantisation** levels. In general, a binary word with bits gives quantisation levels – hence a 3-bit word gives 8 levels, a 4-bit word gives 16 levels, a 5-bit word gives 32 levels, etc.

Figure 18 shows a sine wave along with a set of quantisation levels. Here the 0 or midpoint of the sine wave occurs at a midpoint between the 011 and 100 levels. The gaps between the quantisation levels are called the quantisation intervals. Sometimes a quantisation level is assigned to the 0 or midpoint, which means that on one side of the 0 there is one more level than on the other side.

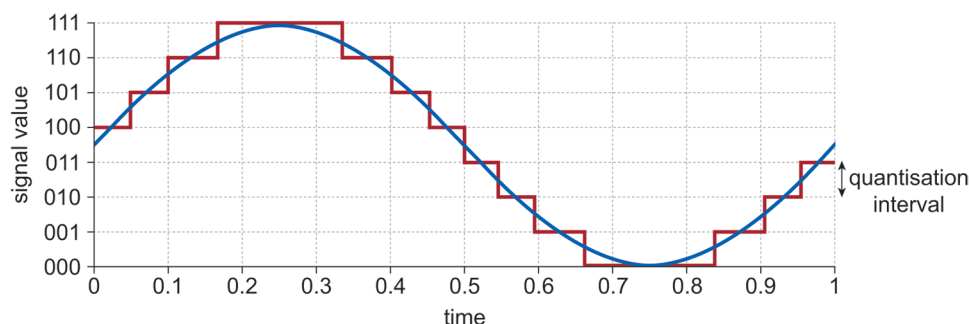


Figure 18 A sine wave and a set of quantisation levels

The larger the quantisation intervals, the more error will be introduced into the sampled signal. The quantisation error for a sample is the difference between the value of the input signal and the resultant quantised signal, with the maximum quantisation error being half the quantisation interval. Quantisation errors can be reduced by increasing the number of levels; however, as the number of levels increases, so does the number of bits needed to represent each sample.

Activity 7

Allow about 15 minutes

An electrocardiogram (ECG) signal contains useful information in frequencies up to 400 Hz.

- At what rate must the signal be sampled to ensure that no information is lost in sampling?
- If a 3-bit quantiser is used to represent the signal, how many bits of data are generated per second if the sampling rate is set to 1000 Hz?
- If the signal range from the ECG extends from +7 V to −7 V, where 7 V equates to the highest quantisation level and −7 V to the lowest quantisation level, what is the quantisation interval and what is the maximum quantisation error in the system?
- If the quantiser is changed to a 4-bit system, what happens to the maximum quantisation error?

Answer

- The signal must be sampled at greater than twice the maximum frequency of the signal, so the minimum sampling rate is 800 Hz.
- With a 3-bit quantiser each sample point uses three bits of data, so the number of bits of data generated is $3 \times 1000 = 3000$ bits per second.
- With a 3-bit quantiser there are a total of eight levels and seven quantisation intervals in the system. If the voltage range covers 14 V, then each quantisation interval is 2 V. The maximum quantisation error is half this quantisation interval, or 1 V.
- A 4-bit system has 16 quantisation levels and 15 quantisation intervals, so each quantisation interval is

This reduces the maximum quantisation error to

In the next section you will return to digital filtering and how they work in the time domain.

3.5 Digital filtering in the time domain

A simple form of digital filter is the three-term averaging filter, in which the output value is equal to the average of three successive signal sample values. In Figure 19(a), which shows a discrete-time signal applied to a digital three-term averaging filter, the output is given by

The signals and are spaced seconds apart in time, where is the sampling interval. Similarly, and are spaced seconds apart.

Figure 19(b) shows the values of the input signal that will be applied to the filter input.

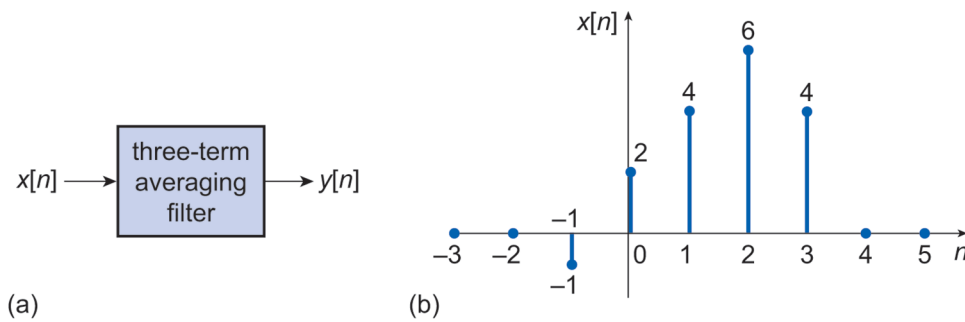


Figure 19 (a) Three-term averaging filter; (b) input signal to the three-term averaging filter

The values of $x[n]$ and $y[n]$ are listed in Table 1. Assume that for any n , $x[n] = 0$ for $n < -3$ and $n > 5$.

Table 1 Values of $x[n]$ and $y[n]$ from Figure 19(b)

n	-3	-2	-1	0	1	2	3	4	5
$x[n]$	0	0	-1	2	4	6	4	0	0

For the output to be calculated, the values of $x[n]$, $x[n-1]$, and $x[n-2]$ must be stored in memory and accessible to the processor performing the calculation. The order of a digital filter is the number of previous inputs (stored in the processor's memory) used to calculate the current output, so this three-term averaging filter is second-order.

For values of n where $n < -1$, the output will be 0, so the calculations below start at $n = -1$.

For $n = -1$, you can substitute in values to give $y[-1] = 0$.

For $n = 0$, you get $y[0] = 0$.

For $n = 1$, you get $y[1] = 0$.

Table 2 shows the results of all calculations for $y[n]$, shown as decimal values to 2 significant figures. Note that you can stop at $n = 5$, since above this the output will be 0 again.

Table 2 Results of calculations for $y[n]$

n	-3	-2	-1	0	1	2	3	4	5
$x[n]$	0	0	-1	2	4	6	4	0	0
$y[n]$	0	0	-0.33	0.33	1.67	4.00	4.67	3.33	1.33

The resulting output discrete-time waveform is given in Figure 20.

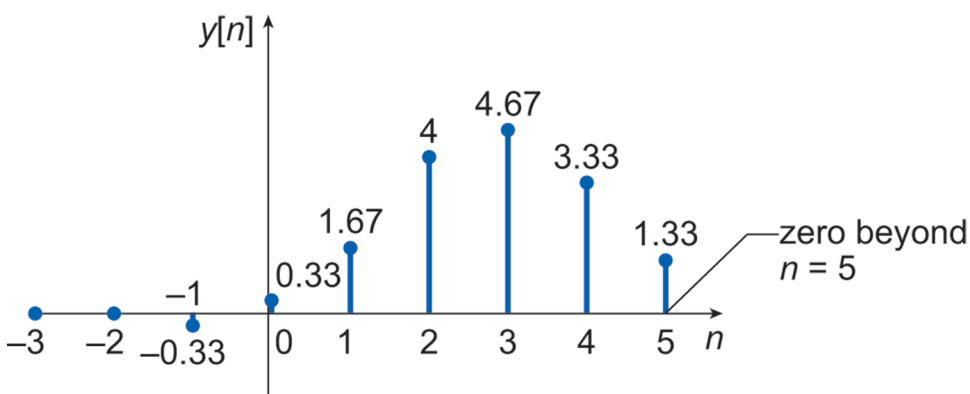


Figure 20 Filter output in response to the input in Figure 19(b)

Figure 21 shows the same filter applied to an input that has more noise in the signal and a longer sequence.

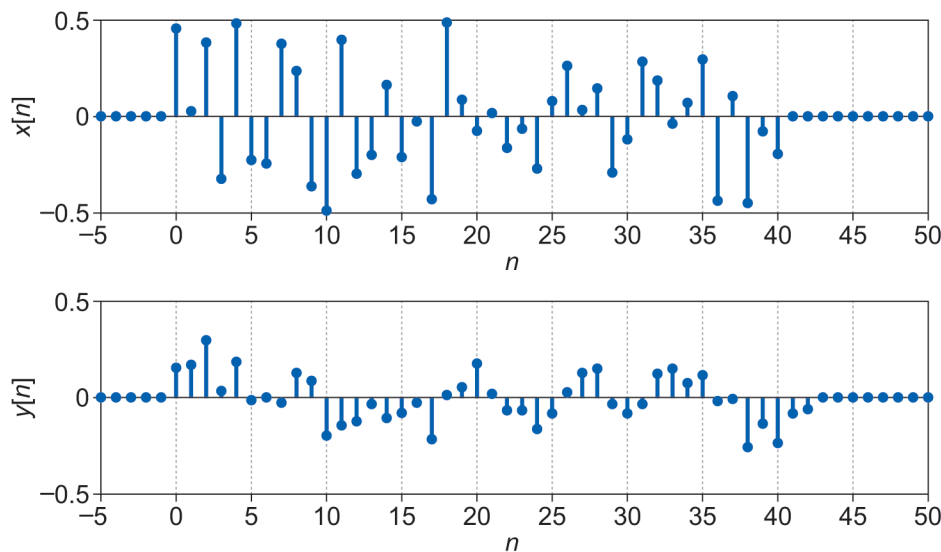


Figure 21 Longer sequence filtered by the three-term averaging filter (Wickert, 2011, p. 7)

Activity 8

Allow about 5 minutes

Looking at Figure 21, describe the effect that the three-term averaging filter has on the output. What kind of filter is this?

Answer

The three-term averaging filter has removed the short-term fluctuations in the signal to show the longer-term trend. This is akin to removing higher-frequency noise from a signal, so it is operating like a low-pass filter.

This digital filter is an example of a system that is both linear and time-invariant, sometimes referred to as an **LTI** system.

You can see that the three-term averaging filter is a low-pass filter, but it is difficult to characterise its response. For example, what frequencies is the filter eliminating from the signal? Earlier in this course, you saw how analogue filters can be designed to give a desired frequency response; now you will look at how digital filters can also be designed in the frequency domain.

Filters are usually described in terms that make sense in the frequency domain, e.g. a low pass filter allows the parts of the signal with low frequencies to pass. In the following section you will see how a digital filter is designed in the frequency domain.

3.6 Designing a digital filter in the frequency domain

Most digital filters are designed in the frequency domain. Input signals are characterised by their frequency spectrum and design filters to modify that spectrum by, for example, removing high-frequency noise with a low-pass filter.

Figure 22 shows the four basic filter structures in the frequency domain. These ideal filters are identical for both analogue and digital filters (you have already seen them earlier in the course in Section 2.2).

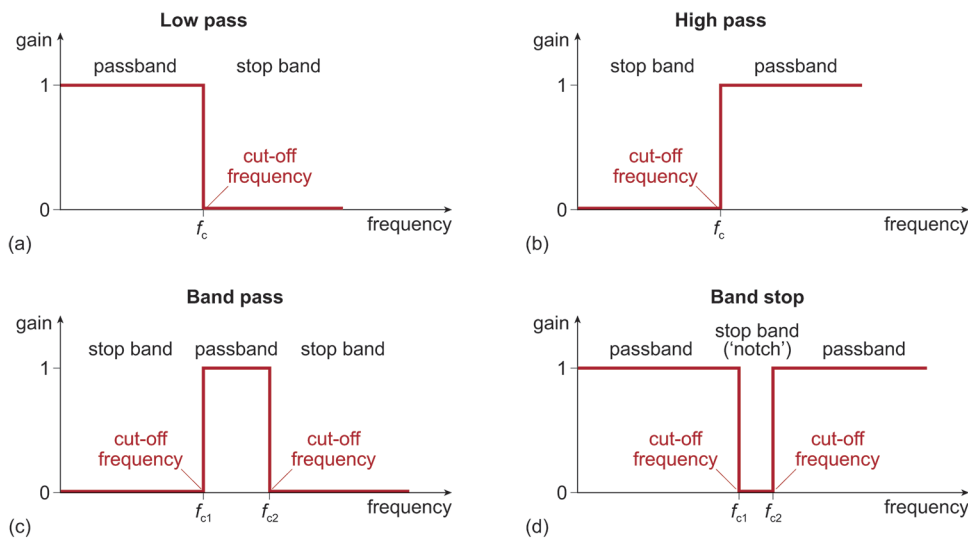


Figure 22 Ideal filter responses: (a) low-pass; (b) high-pass; (c) band-pass; (d) band-stop (repeat of Figure 5)

In the design process, the aim is to produce a filter frequency response that best matches the profile of the filter; however, compromises have to be made. You have already seen how the order of the design of the filter affects the roll-off, so decisions are made in the design of these analogue filters to best match the filter to the ideal response.

Figure 23 shows some of the characteristics of a typical low-pass digital filter. In comparison to the ideal shape (shown in red), there is a transition region between the passband and the stop-band sections, and also a ripple in both the passband and the stop band. These effects can be altered by changing various parameters in the design.

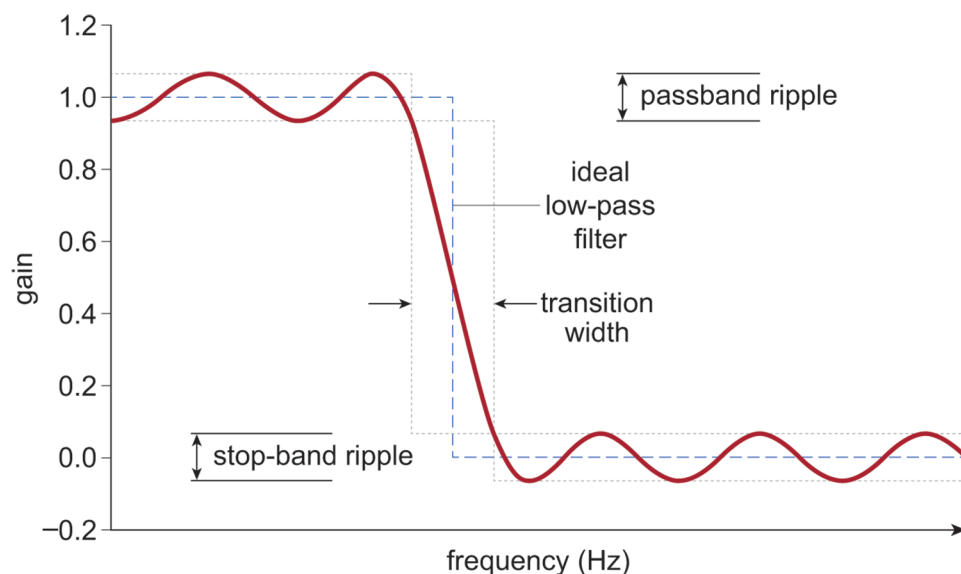


Figure 23 Typical low-pass digital filter response

When designing a digital filter in the frequency domain, software tools are most often used to generate the mathematical expression for the filter. This expression is in the form of a

difference equation – an equation involving combinations of samples at specific times. You have already seen a difference equation in this section: the equation $y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$ for the three-term averaging filter. As another example, the following difference equation contains five terms:

In a digital filter design, the number of terms in the difference equation is often referred to as the number of **taps**, and is specified as a design parameter. A larger number of taps may give a filter design that more closely matches the desired specification; however, more taps will mean that it takes longer to compute the filter outputs.

There are two main classes of LTI (linear time-invariant) digital filter: the finite impulse response (FIR) filter and the infinite impulse response (IIR) filter. An IIR filter requires fewer computations to achieve the same performance as an FIR filter, so has a speed advantage. However, an IIR can have stability issues and also non-linear phase issues (where signals of different frequencies are delayed by different amounts, resulting in a distortion of the output signal). The difference between an FIR filter and an IIR filter is that the FIR filter uses only the filter inputs when generating its output, whereas an IIR filter uses both the filter inputs and past filter outputs – in other words, it uses feedback. The difference equation above is for an FIR filter. The difference equation below has four terms and the final term is an output value, so it is an IIR filter:

To calculate the output of an IIR filter, both previous input samples and previous output samples are stored in the processor's memory. The order of the IIR filter is the larger of the number of input values stored and the number of output values stored. In the above example, the filter is second-order, since two previous input values are stored but only one previous output value.

Before you look at a filter being designed, you need to know a little more about the relationship between the time domain and the frequency domain. You will cover this next.

3.7 Fourier transforms and the sinc pulse

You saw earlier (Figure 5) that the ideal frequency responses shown in Figure 22 are sometimes referred to as brick-wall filters because of the sharp transitions between passbands and stop bands. In other words, they are rectangular functions. However, whilst it is possible to use a rectangular function in the frequency domain to specify the filter, you must perform the calculations to implement the filter in the time domain. To do this, you need to translate between the time and frequency domains; in particular, for a brick-wall filter, you need to know what a rectangular function in the frequency domain looks like in the time domain. For a continuous-time system, you would use a **Fourier transform** to do this; for a discrete-time system, you use a corresponding discrete-time Fourier transform.

You can perform mathematical calculations on paper to work out the Fourier transform of a signal in either the time or the frequency domain. Under those circumstances, you would use the formula for either the discrete or the continuous transform, depending on the type of system you were dealing with. However, the majority of Fourier transforms will be carried out by a computer system – even if the system is dealing with continuous-time signals as input and output, the signal processing will be happening in the discrete-time domain of the computer. There is a particular algorithm called the fast Fourier transform

(FFT) that is used to carry out Fourier transforms efficiently. Such was the need to perform these calculations at great speed that the FFT was included in a list of the top 10 algorithms of the twentieth century by the IEEE journal *Computing in Science & Engineering* in the year 2000 (Dongarra and Sullivan, 2000).

Using the discrete-time Fourier transform, you can see that the time-domain representation of a rectangular function in the frequency domain is the sinc pulse, as shown in Figure 24.

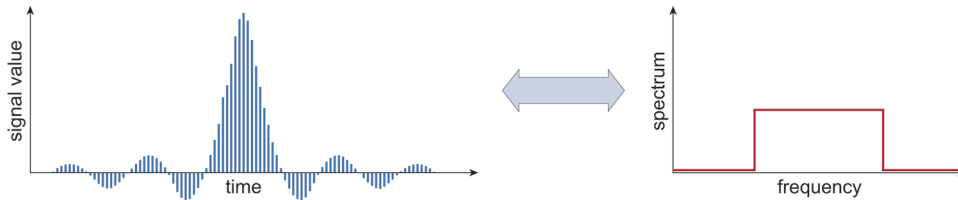


Figure 24 Fourier transform pair: a rectangular function in the frequency domain is represented as a sinc pulse in the time domain

Mathematically, a sinc pulse or sinc function is defined as $\sin(x)/x$.

Figure 25(a) and Figure 25(b) show a sinc envelope producing an ideal low-pass frequency response. However, there is an issue because the sinc pulse continues to both positive and negative infinity along the time axis. Whilst mathematically you can readily take the Fourier transform of a sinc pulse, it can't be computed because of the extension to infinity. The obvious solution is to truncate the sinc response as in Figure 25(c), so that the ripples no longer extend to infinity. Now that the pulse is finite, it can be shifted so that it only has positive sample numbers. The effects of this in the frequency domain are shown in Figure 25(d) – ripples in the passband and the stop band. In essence this shows why you can never have the perfect ideal or 'brick-wall' filter.

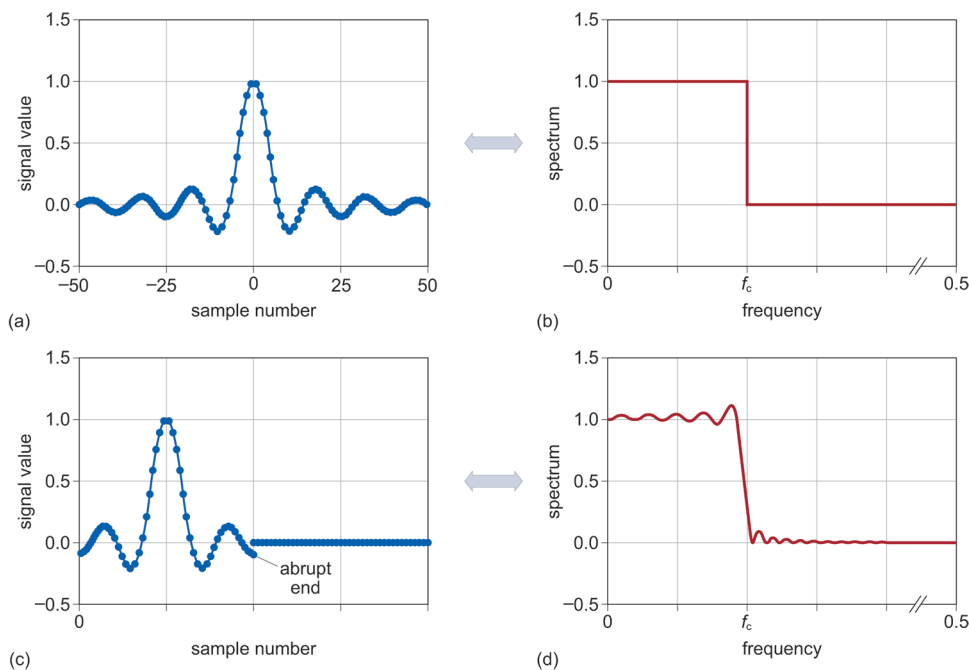


Figure 25 (a) Ideal sinc function in the time domain; (b) frequency response of the ideal sinc function; (c) abruptly truncated sinc function in the time domain; (d) frequency response of the truncated sinc function

A technique for dealing with the truncated sinc is to apply a **window function** that brings the endpoints of the truncated sinc to zero. Figure 26(a) shows a suitable shape of window, Figure 26(b) shows the effects of applying the window to the truncated sinc and Figure 26(c) shows the resultant frequency response.

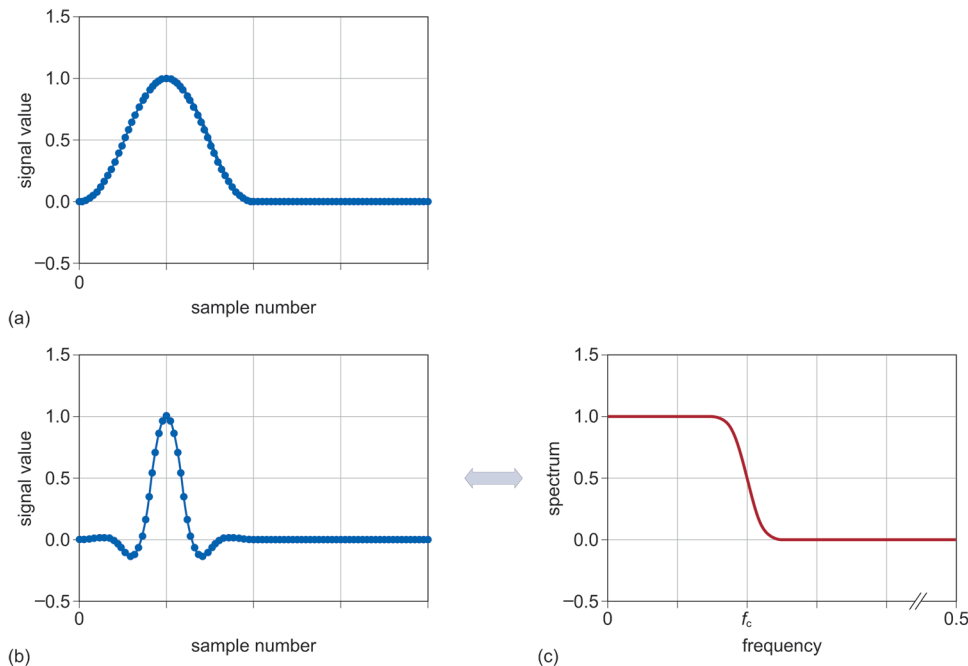


Figure 26 Using windowing to compensate for a truncated sinc pulse: (a) suitable window shape; (b) window applied to the truncated sinc; (c) resultant frequency response

When a window function is applied, it is effectively ‘multiplied’ with the sinc function. The process used to do this is called **convolution**. Convolution is outside the scope of this course, but when applied, what is left is where the signals overlap. You can think of this as a ‘view’ through the window function.

There are various shapes of window that can be applied. Two common shapes are the Blackman window and the Hamming window, although there are others (such as Kaiser, Bartlett and Hann). The Hamming window gives a better transition response, but the Blackman window has better stop-band gain and lower passband ripple.

You are now ready to design a digital filter and try it out.

3.8 A low-pass filter design

You will now look at a real example of low-pass filter design. For this, a software package called Signal Wizard has been used. You are not required to download Signal Wizard, but for information, it is a free package that can be installed on your computer. Details of the Signal Wizard and where to find it are given can be found here: [Installing Signal Wizard for use in offline mode](#).

Figure 27 shows a screenshot from Signal Wizard which shows the input signal waveform both in the time domain (the ‘Time Waveform’) and in the frequency domain (the ‘Frequency Spectrum’). The aim is to remove any frequencies in this waveform above 5000 Hz, so a low-pass filter with a 5000 Hz cut-off frequency is required.

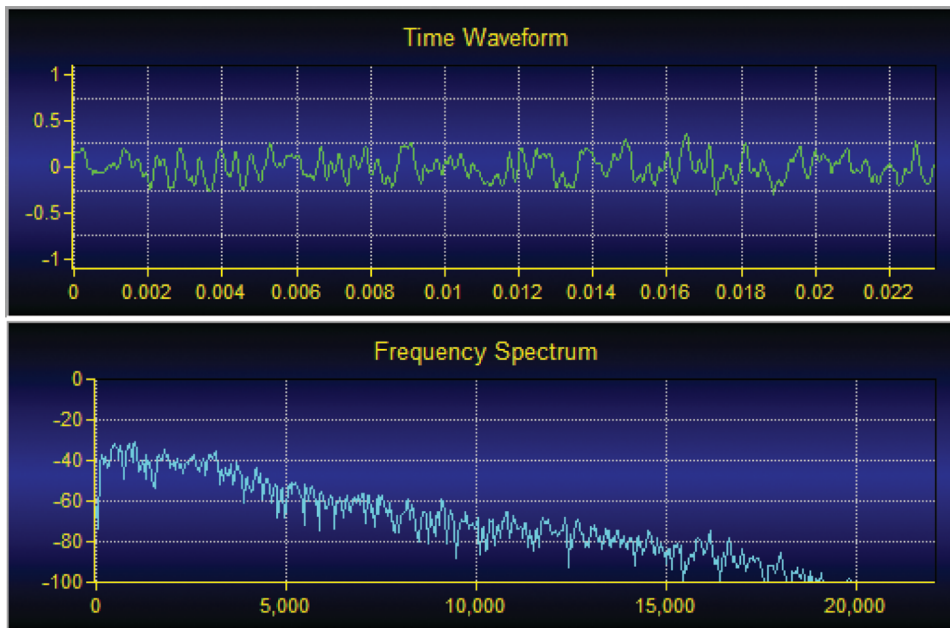
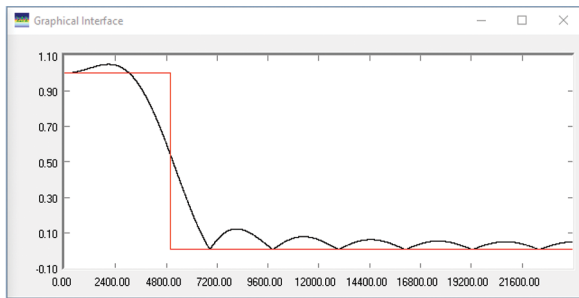
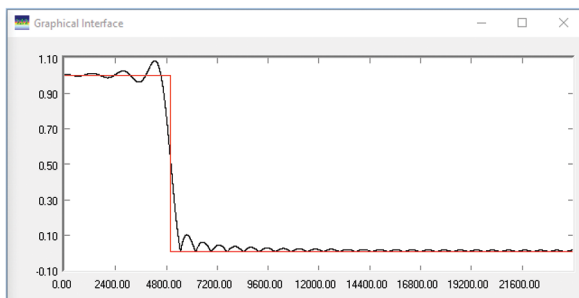


Figure 27 Characteristics of the input signal

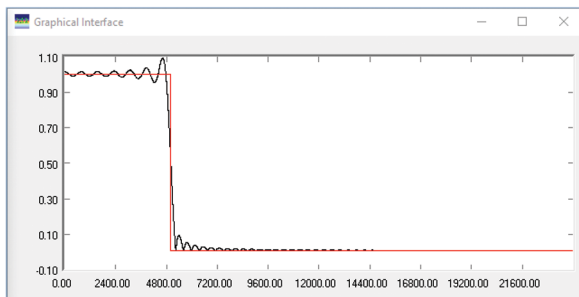
The specification of an FIR low-pass filter with a gain of 1 and a cut-off frequency of 5000 Hz is entered into the filter design interface. The graphical interface windows in Figure 28 all show the 'brick-wall' specification in red with the implementation in black. These designs all use a rectangular window, which gives an abruptly truncated sinc function. The first design uses 15 taps (Figure 28(a)), the second uses 63 taps (Figure 28(b)) and the third uses 127 taps (Figure 28(c)). All other parameters are unchanged. As the number of taps in the design increases from the top image to the bottom, the transition zone narrows and the designed filter more closely matches the filter specification. However, the amplitude of the ripples in the passband and the stop band remains unchanged, although the frequency increases as the number of taps increases.



(a)



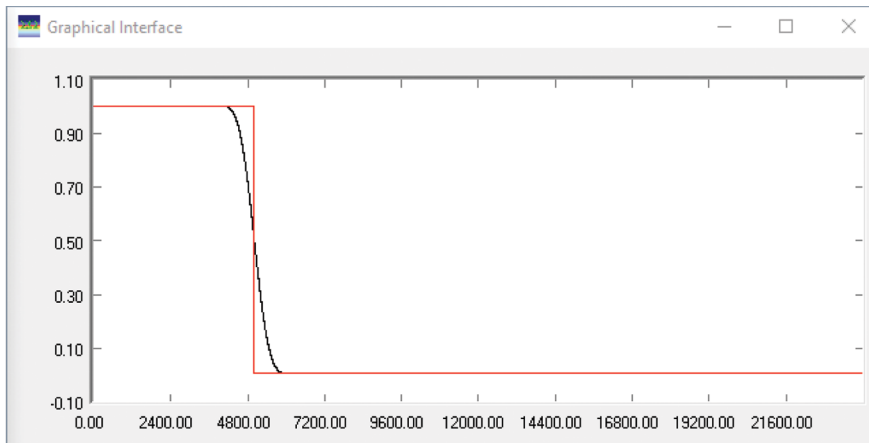
(b)



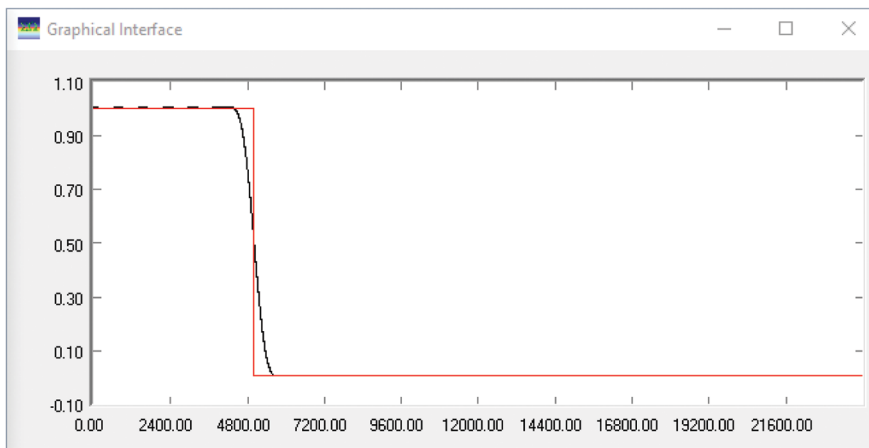
(c)

Figure 28 Digital filter design with a rectangular window: (a) 15 taps; (b) 63 taps; (c) 127 taps

The next set of designs, in Figure 29, keeps the number of taps at 127 and varies the window function used. The design implemented in Figure 29(a) uses a Blackman window, while the design implemented in Figure 29(b) uses a Hamming window. You can see that both have reduced the ripples in the passband.



(a)



(b)

Figure 29 Filter designs with different window functions: (a) 127 taps with Blackman window; (b) 127 taps with Hamming window

Figure 30 zooms in on sections of the transition band and passband to see the effects in more detail. Comparing Figure 30(a) and Figure 30(c) shows that the transition zone of the Hamming window is narrower than that of the Blackman window. Comparing Figure 30(b) and Figure 30(d) shows that the Blackman window has reduced the ripples in the passband more than the Hamming window. These results confirm that the Hamming window gives a better transition response, while the Blackman window has lower passband ripple.

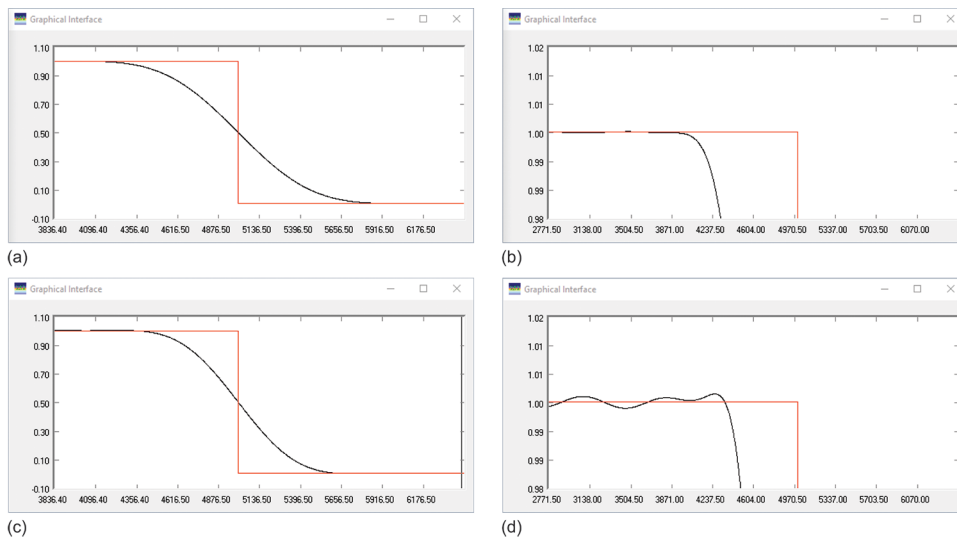


Figure 30 Details of the Blackman and Hamming windows: (a) 127 taps with Blackman window, transition zone; (b) 127 taps with Blackman window, passband; (c) 127 taps with Hamming window, transition zone; (d) 127 taps with Hamming window, passband

The input signal was filtered using the 127-tap Hamming window design, and the output response is shown in the 'Time Waveform' and 'Frequency Spectrum' charts in Figure 31. The frequency spectrum shows that the filter has indeed taken out the redundant higher-order frequencies above 5000 Hz.

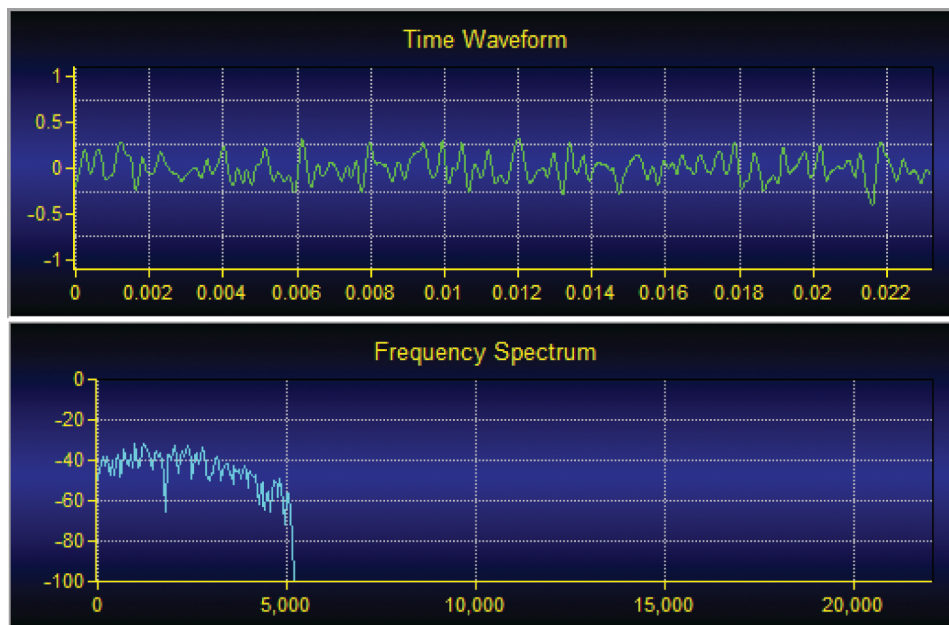


Figure 31 Filter response output

When a digital filter such as the one above is being implemented, the mathematical calculations will be defined in a software program, which will then run on some digital hardware. The basic processes taking place in the digital hardware are adding and subtracting; these processes will occur thousands, probably millions of times in the filtering of a sampled signal. **Digital signal processor (DSP)** chips are designed specifically to carry out these calculations, and their internal structure (referred to as their architecture) has been optimised to do so – thus it is different from that of a general-

purpose processor used in a computer. There are several major DSP chip manufacturers, including Texas Instruments and Analog Devices.

You've therefore seen that usually a digital filter is designed using a software package. To finish this course, you will have a chance to explore digital filtering using an interactive resource. The next section introduces this resource.

3.9 Digital filtering in practice

You are now going to use an interactive resource to add Gaussian noise to a noise-free (or 'clean') data signal. The noisy data signal is passed to a detector, which determines whether received samples are binary zeros or ones. The added noise causes the detector to make mistakes; hence there are bit errors. Using a finite impulse response (FIR) digital filter, you will 'clean up' the noisy data signal and thereby reduce the bit-error rate.

Right-click on the image or link below to open Interactive 1 in a new tab or window so you can continue to work through the course and activities with the interactive alongside. The interface consists of four equally sized zones: top left, top right, bottom left and bottom right. Note that depending on the browser you are using, the sliders and other components in the interface may have a different visual appearance from those shown in the screenshots in this document, but the functionality will be the same.

Interactive content is not available in this format.

[Interactive 1 FIR filter](#)

The top left zone shows the signal, whether it is clean or noisy. When the interactive resource is launched, the noise level is at zero and the signal is clean. The data signal is shown as a single step from 0 to 1, but there are 10 000 samples for this data signal, half of which are from the 0 part of the signal and half of which are from the 1 part. Noise is added by sliding the **Noise level (V)** slider to the right and clicking on **Apply**.

The top right zone displays a histogram of the signal. As there is an equal number of zeros and ones in the signal, the probability of each is 0.5. The detected probabilities of zeros and ones in the received signal are shown by the vertical red lines situated at 0 and 1. Without added noise, perfect detection can be achieved.

The top right zone allows the decision level at the detector to be set using the **Decision level (V)** slider, and also the degree of low-pass filtering to be set using the **FIR filter width (number of samples)** slider. Finally, the top right zone also gives statistics about the number of true and false detections. Figure 32 explains how to interpret this data. Before noise is added, all samples are correctly identified, so **False detections** should be at 0 to start with.

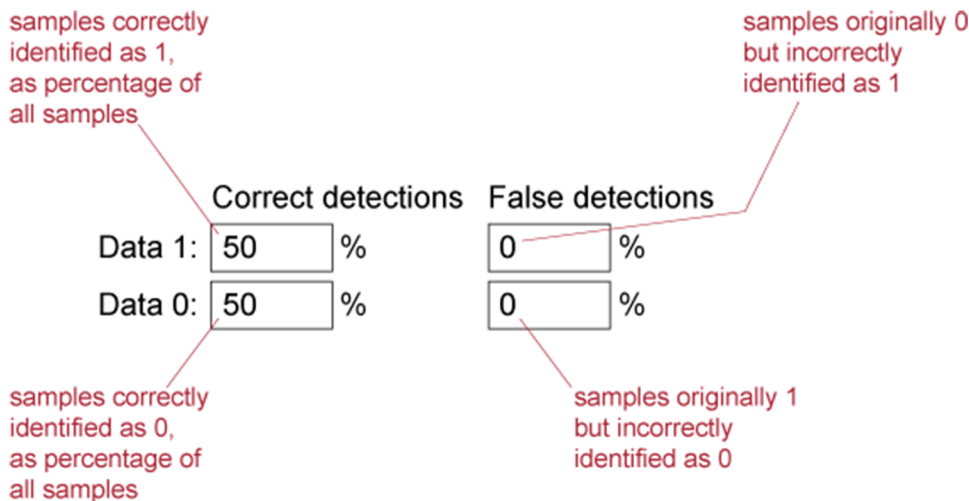


Figure 32 Interpreting the statistics

Note in Figure 32 that if the correct detection of data 1s were to drop below 50%, the deficit would appear as an increase in the false detection of data 0s. Similarly, if the correct detection of data 0s drops below 50%, the deficit should appear as an increase in the false detection of data 1s. Thus the diagonal pairs of statistics in Figure 32 should always add up to 50% – or approximately 50%, given the possibility of rounding errors in the calculation.

The bottom left zone shows the effect of filtering applied to a noisy signal. The bottom right zone is similar to the top right zone, showing a signal histogram and false detection statistics; however, these are after filtering, whereas the top right zone is before filtering. In the next section you will have a go at adding noise to the data signal in the interactive.

3.10 Adding noise

After familiarising yourself with the interface in Interactive 1, try adding noise in the top left zone. The **Noise level (V)** slider is not calibrated, except for having a minimum value of 0 and a maximum of 1. Judging by eye, set the slider to around 0.1 V and click on **Apply**. You can hear what the noisy signal sounds like by going to **Audible signal with noise** and clicking on the **Play** (triangle) icon. The top part of the interface should resemble Figure 33.

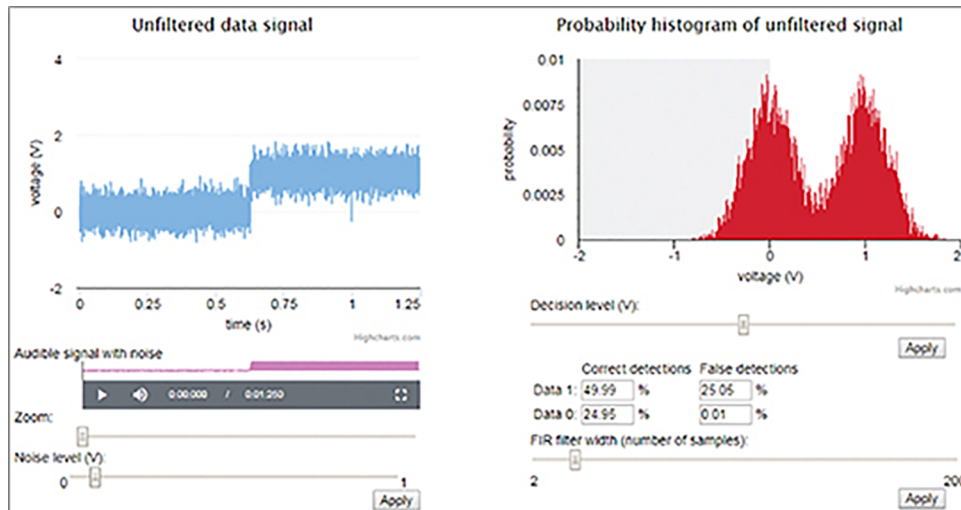


Figure 33 Noise applied to the data signal

Activity 9

Allow about 5 minutes

Why has the display in the top right zone changed? What is the display now showing?

Answer

The effect of the added noise is to change the voltages representing 0 and 1 randomly around the mean values of 0 and 1. The display shows a histogram of the probabilities of particular signal voltages. Voltages close to the mean are most likely, so the histogram has peaks at 0 and 1; voltages further away are possible, but less likely the further you go from the mean values.

Next you will look at changing the decision level.

3.11 Changing the decision level

Again, using the interactive in Section 3.9 you will now look at changing the decision level.

The decision level is represented by the right-hand vertical edge of the grey area on the histogram display. Use the **Decision level (V)** slider to set the decision level midway between 0 and 1. You will have to judge its position by eye. Click on **Apply** to implement the decision level and look at the statistics.

Activity 10

Allow about 5 minutes

- Why are there false detections?
- Why are the detection rates (correct and false) practically the same for 0 and 1?

Answer

- The effect of noise is to take the signal occasionally over the decision level, so that a 0 is detected as a 1 and vice versa.

- b. The percentages of correct and false detections are virtually the same for 0s and 1s because of the symmetry of the situation. The noise affects the 0s and 1s equally, and the decision level is symmetrically placed between 0 and 1.

The symmetrical arrangement that you have used so far, with the decision level halfway between the two binary symbols, is typical of much practical implementation of binary signal detection, but looking at an asymmetrical arrangement is instructive.

Use the **Decision level (V)** slider to place the decision level asymmetrically between 0 and 1 (that is, much closer to 1 than to 0, or vice versa), then click on **Apply**.

Activity 11

Allow about 10 minutes

Explain, in general terms, the correct and false detection statistics that have resulted from your asymmetrical placement. You will not be able to give a precise account, but you might be able to explain the relative sizes of the statistics.

Answer

You may have got something similar to Figure 34, with the decision level fairly close to 1.

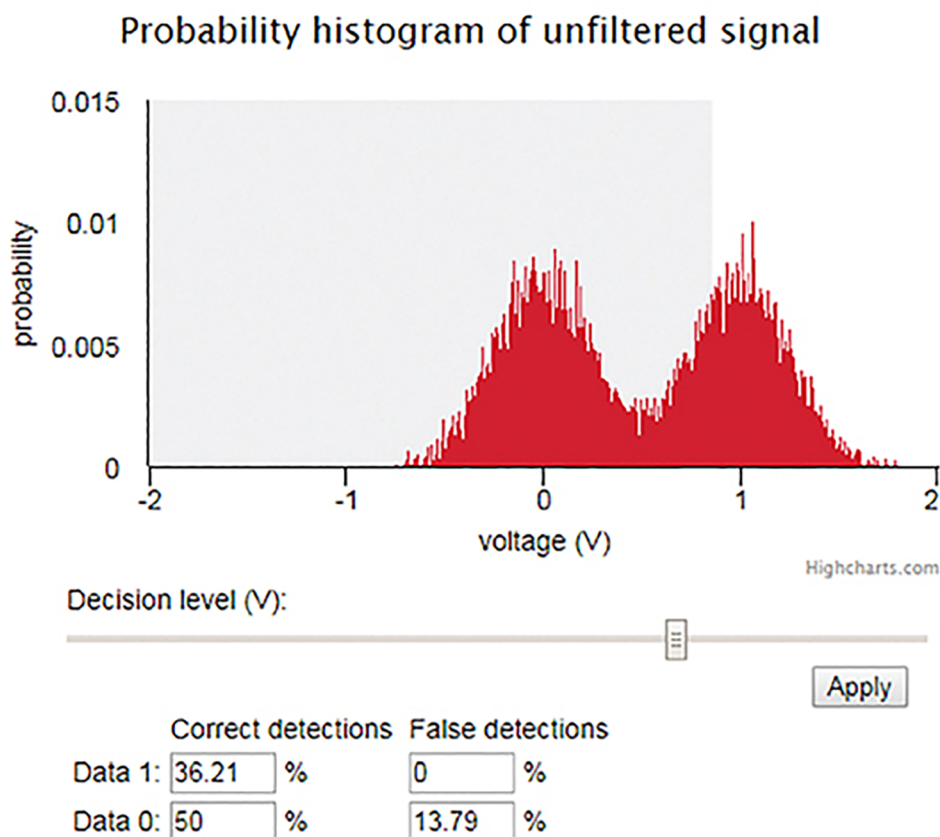


Figure 34 Asymmetrical decision level

Because the decision level is so close to 1, noise-affected 0 signals very rarely go beyond the decision level. This is why false detections of 1 are at 0%, as this statistic reflects 0s that are wrongly detected as 1s. As there are no false detections of 0s, all detections of 0 must be correct, which is why correct detections of 0 are shown at 50%. With the decision level close to 1, noise-affected 1s quite often drop below the decision level and are detected as 0s. This is why the false detection rate for 0 is relatively high, at almost 14%. Correspondingly, the correct detection rate for 1 is relatively low, at just above 36%. These two statistics add to approximately 50%.

Now you have explored how adding noise and changing the decision level affects the data signal, to end this course you will apply the FIR filter.

3.12 Applying the FIR filter

Use the **Decision level (V)** slider to return the decision level to midway between 0 and 1, then click on **Apply**. Confirm that the statistics are as would be expected; that is, correct detections of 1s and 0s are at approximately the same percentage. The correct detection percentages do not need to be identical, but try to get them to within about 1% of each other by adjusting the detection level.

You are now going to apply the FIR filter. In the top right zone, set the **FIR filter width (number of samples)** slider by eye to around 30–40 samples, then click on **Apply**.

Activity 12

Allow about 5 minutes

- Compare the top right zone (histogram and statistics before filtering) to the bottom right zone (histogram and statistics after filtering). What effect has the filter had on the statistics for correct detection?
- Explain any difference in detection statistics between those beneath the unfiltered signal and those beneath the filtered signal.

Answer

- The filter increases the correct detection rate.
- The filter reduces the spread of values around the mean, which is shown by the fact that the histogram peaks are narrower than in the unfiltered histogram. The reduced spread of values reduces the probability of sample values crossing the decision level.

Some points to consider as a result of your work with the FIR filter interactive resource are as follows:

- Noise causes the actual voltages for 1s and 0s to be distributed around the intended voltage.
- The distribution of voltages can cause errors when, for example, the voltage representing a 1 is closer to 0 than to 1.

- The wider the distribution (that is, the noisier the signal), the more likely errors become.
- Filtering can reduce the width of the distribution of voltages, thereby reducing the error rate.

Conclusion

In this course you have learned about discrete-time signals and the discrete-time systems that use them. In doing so, you have focused on digital filtering and found out why advances in digital computer processing have allowed digital filtering to be used in scenarios where analogue filters would once have been the only viable solution. You have seen how relatively simple averaging filters can remove high-frequency noise, and also that more complex filters are designed.

This completes your study. You should now be able to:

- understand how filtering of discrete-time signals can be achieved by mathematical processes such as averaging
- understand how mathematical operations applied to a discrete-time signal in the time domain can result in the removal or reduction of unwanted aspects of the signal
- explain why filters are designed in the frequency domain, and specify a digital filter to achieve a desired filtering effect.

This OpenLearn course is an adapted extract from the Open University course [*T312 Electronics: signal processing, control and communications*](#)

Glossary

alias

An error appearing in a sampled signal when the bandwidth of the signal is greater than half the sampling frequency (that is, when the sampling frequency is lower than the Nyquist frequency). Such effects are also referred to as artefacts or ghosts.

anti-aliasing filter

A low-pass filter that is able to remove aliasing in sampled signals by cutting all the spectral components that are greater than or equal to half the sampling frequency.

Bode plot

Loosely, a graph of the frequency response of a device or system. Strictly, a pair of graphs showing frequency response and phase response over the same span of frequencies.

convolution

A mathematical operation that combines two signals to produce a third signal. When two signals are convolved, the resultant third signal expresses how the shape of one signal is modified by the other.

decibel

A logarithmic way of expressing a power ratio. For powers P_1 and P_2 , their ratio in decibels is defined as $10 \log_{10} (P_1/P_2)$. The symbol for decibels is dB. Strictly the decibel is not a unit, as any ratio must be a pure (that is, dimensionless) number. However, it is often regarded as a unit.

difference equation

An equation in which all variables have been sampled at fixed intervals, and these variables are multiplied by some coefficient.

differential equation

A mathematical equation in which one or more terms contains a mathematically differentiated variable.

digital signal processor (DSP)

A semiconductor device similar to a microprocessor. Whilst a microprocessor is a general-purpose device, a DSP has been optimised to carry out the computations used for processing discrete signals.

first-order

As applied to a filter, the simplest type of filter, having in its passive form a single reactive element (a capacitor or an inductor) and a roll-off of 20 dB/decade, or 6 dB/octave. As applied to a differential equation, such an equation in which the main variable is differentiated once. Any system that can be modelled with such a differential equation would be referred to as a first-order system.

Fourier transform

A transformation that extends the concept of the Fourier series to non-periodic signals. It allows us to estimate the spectrum of a signal and perform a frequency analysis.

frequency response

The response of a system (e.g. a filter) when we input sine waves at different frequencies (but equal amplitude). It tells us how the system will modify the spectrum of any input signal we feed to the system.

gain

In amplification, a measure of how many times the input signal amplitude is increased. It is generally measured as the ratio between the input signal amplitude and the output signal level. If a gain value is given as just a number (i.e. with no units), the gain is likely to be a ratio of voltages; if the value is given in decibels, it is a ratio of powers. See also voltage gain and power gain.

octave

The span of frequencies covered by a doubling of frequency, or by a halving of frequency. For example, the frequency span from 500 Hz to 1000 Hz is an octave, as is the span from 500 Hz to 250 Hz. In music, the eight notes of a diatonic scale (that is, doh, re, me ... ti, doh) cover an octave; hence the name 'octave' for this span of frequencies.

operational amplifier

A general-purpose analogue amplifier intended to be used as a component in other electronic circuits, and usually supplied as a multi-pin integrated-circuit device with two inputs and a single output. Typically an op-amp is a differential amplifier (that is, it amplifies the difference between its two inputs) and has an unusably high gain and extremely high input impedance. To give useful and predictable behaviour, external feedback circuitry must be applied. This circuitry determines essential parameters such as input impedance, output impedance, overall gain and frequency response, and also whether the circuit operates as a single-input amplifier or a differential amplifier.

order

A numerical classification for filters (e.g. 'first order', 'second order', 'third order', etc.). The order is determined by the differential equation of the filter. For a first-order filter, the highest differential coefficient in the equation is first-order (e.g. dv/dt); for a second-order filter, the highest differential coefficient is second-order (e.g. d^2v/dt^2). The higher the order, the steeper the roll-off and the sharper the cut-off between passband and stop band. Increasing the order by one adds 20 dB/decade to the filter's roll-off.

passband

The band or bands of frequencies passed by a filter with least attenuation, or no attenuation. Frequencies outside the passband are cut off, or stopped, by the filter. Passband is the counterpart of stop band.

power gain

The ratio of output power to input power. It is usually expressed in decibels (dB). A power gain of 0 dB means that the output power is the same as the input power. A power gain of 3 dB (or, more exactly, 3.0103 dB) means that the output power is double the input power.

quantisation

Conversion of an analogue quantity, which could take any value within a range, to one of a set of discrete values.

roll-off

The steepness of a filter's attenuation in a stop band. Also, the steepness of the attenuation of any device that produces attenuation (for example, a linear amplifier at the extremes of its operating-frequency range).

stop band

The band or bands of frequencies stopped, or cut off, by a filter. The counterpart of the passband.

taps

In a digital filter, the number of taps is the number of terms in the mathematical expression that describes the filter. This expression is given in the form of a difference equation. In digital filter design, the maximum number of taps to be used in the implementation is required as part of the design specification.

voltage gain

For a sinusoidal input and output, voltage gain is the ratio of the output voltage's amplitude to that of the input voltage. It has no units.

window function

In signal processing, a mathematical function that is zero-valued outside some chosen interval. When a signal is convolved with a window function, the resultant signal is also zero-valued outside the chosen interval, so it is the original signal viewed through the window function.

References

- Asgari, S. and Mehrnia, A. (2017) 'A novel low-complexity digital filter design for wearable ECG devices', *PLOS ONE*, vol. 12, no. 4 [Online]. Available at <https://doi.org/10.1371/journal.pone.0175139> (Accessed 25 March 2019).
- Dongarra, J. and Sullivan, F. (2000) 'Guest editors' introduction: the top 10 algorithms', *Computing in Science & Engineering*, vol. 2, no. 1, pp. 22–3.
- Wickert, M. (2011) 'Chapter 5 FIR Filters', *ECE 2610 Introduction to Signals and Systems* [Online], University of Colorado Colorado Springs. Available at www.eas.uccs.edu/~mwickert/ece2610/lecture_notes/ece2610_chap5.pdf (Accessed 3 June 2019).

Acknowledgements

This free course was written by Allan Jones, Bernie Clarke and Phil Picton. It was first published in July 2020.

Except for third party materials and otherwise stated (see [terms and conditions](#)), this content is made available under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 Licence](#).

The material acknowledged below is Proprietary and used under licence (not subject to Creative Commons Licence). Grateful acknowledgement is made to the following sources for permission to reproduce material in this free course:

Figures

Course image: paulclee / www.pixabay.com

Figure 1: © BBC

Figure 2: © NASA

Figure 3: Courtesy Starship Technologies

Figure 14: Adapted from: <https://doi.org/10.1371/journal.pone.0175139>

Installing Signal Wizard for use in offline mode PDF, Figure 1: Taken from: <http://www.signalwizardsystems.com/>

Every effort has been made to contact copyright owners. If any have been inadvertently overlooked, the publishers will be pleased to make the necessary arrangements at the first opportunity.

Don't miss out

If reading this text has inspired you to learn more, you may be interested in joining the millions of people who discover our free learning resources and qualifications by visiting The Open University – www.open.edu/openlearn/free-courses.