# **Signals**, Systems and Control

## Contents

## 1.6 Discrete time frequency transformations and spectra

### 1.6.1 More Time - Frequency domains

The domains we have encountered so far include continuous time; discrete time; and continuous frequency (Fourier) and continuous complex frequency (Laplace). The fact that I have now started adding 'continuous' into the description of the frequency domain should give you a clue as to what we are going to introduce next!

The table below introduces two new domains – 1) Discrete frequency normalised to sequence length, k, This is useful for the practical transforms we are going to encounter in this lecture; 2) Discrete complex frequency, or the Z domain is the discrete ('digital') version of the Laplace domain and its use is implicit if we are modelling systems in a microcontroller or other digital systems. We won't cover it in this lecture series (we can't fit everything in), but you should be aware of it and the theory we do cover in this unit will give you a good foundation if you need to use the Z domain in future.

| **Time Domains** | **Frequency Domains** |
|---|---|
| Continuous time, $t$ | Continuous frequency, $\omega$, |
| Discrete time, $n$ | Continuous complex frequency, $s$ |
| | Continuous frequency normalised to sampling period, $\Omega$ |
| | Discrete frequency, normalised to sequence length, $k$ |
| | Discrete complex frequency, $z$ |

### 1.6.2 Discrete domains – why do we have them?

You will recall the discussion about the need for discrete time signals some lectures back: the argument for discrete domains derives from the fact that digital systems cannot have an infinite number of samples or allow those samples to take an infinite number of values. This applies as much in the frequency domain as it does in the time domain – thus we can only have a limited number of frequency components (and they can only have a fixed number of magnitudes).

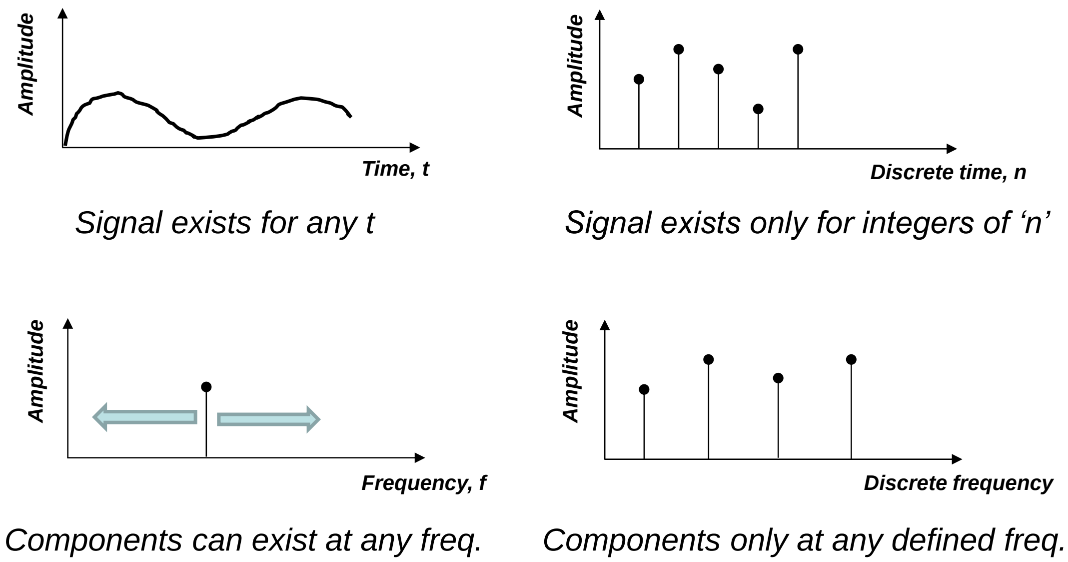



***Fig 1. The frequency domain can be discretised similar to the time domain – in which case components only exist at defined frequencies***

### 1.6.3 Discrete time Fourier transform - DTFT

The Fourier and Laplace transforms we encountered in lecture 1.5 were continuous in both time and frequency domains. We are now going to think about transforms where we discretise one or both of these domains, starting with the Discrete Time Fourier Transform (DTFT).

The DTFT takes a discrete time domain signal and maps it the normalised frequency domain Ω (you will recall Ω is the signal frequency normalised to the sampling frequency). The key thing to remember here is that although Ω is a normalised domain, it is continuous.

The DTFT:

$$X(\Omega) = \sum_{n=-\infty}^{n=\infty} x[n]\, e^{-j\Omega n}$$

*n = current sample*

*Ω = normalised frequency, radians per sample. $0 \leq \Omega < 2\pi$*

***Discrete time so correlation integral becomes a summation***

***Phasor at Ω***

***Discrete time series***

Because it is sampled, the continuous spectrum is still subject to aliasing, i.e. Ω is only unique over the range $0 \leq \Omega < 2\pi$

The inverse DTFT reverts back to integration because the frequency domain is continuous:

$$x[n] = \frac{1}{2\pi} \int X(\Omega)\, e^{j\Omega n} d\Omega$$

What else might you notice with the DTFT? Well the summation is over -∞<n<∞, which might seem odd since we have previously said we start a discrete sequence at n=0; you can think of this as being equivalent to a continuous time signal where we might in practice have a signal that starts now (t=0) and extends into the future, but mathematically we might describe it for all time – the maths is not aware that we have attached the physical concept of time to a particular domain and we can't go back in time!

You don't have to learn the DTFT but it is useful for the next section if you can follow the discussion in this section.

### 1.6.4 Discrete Fourier transform - DFT

The DTFT was just a steppingstone to get to where we really want to be – a transform that is discrete in both time and frequency domains – such a transform can be implemented in our digital systems.

To do this we just need to choose a fixed number of phasors of different frequencies to correlate with – the trick is to choose these carefully! We will also start to consider a sampled sequence of finite length – this will impact the minimum discrete frequency we can resolve the time sequence into. All this can appear as a bit of a brain-teaser, but persevere and it will make sense.

The Discrete Fourier Transform (DFT) is given by:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N}$$

*N = total number of samples in sequence*

*k = cycles per sequence length*

*k only takes integer values 0≤k<N-1*

To better understand what is happening in the DFT, let's compare with the DTFT and the FT to see how we get from one to the other.

| FT (frequency, Hz) | DTFT | DFT | notes |
|---|---|---|---|
| $X(f)$ | $X(\Omega)$ | $X[k]$ | The DTFT is a continuous function (round brackets) over the continuous range Ω; the DFT is a discrete function (square brackets) over the discrete range k, i.e. k only takes certain values. |
| $\int_{-\infty}^{\infty}$ | $\sum_{n=-\infty}^{n=\infty}$ | $\sum_{n=0}^{N-1}$ | The correlation in the DTFT is performed over all of discrete time; in the DFT the correlation is performed over a limited period of discrete time corresponding to the length of the data i.e. the data is 'N' samples long. Note the first sample is at n=0, thus the last is at n=N-1. |
| $x(t)$ | $x[n]$ | $x[n]$ | The DTFT and DFT are made up of discrete time domain sequences, hence square brackets. |
| $e^{-j2\pi ft}$ | $e^{-j\Omega n}$ | $e^{-j2\pi kn/N}$ | This is where the interesting things happen! Recall that this part specifies the phasor we correlate with the time domain signal, which in turn determines which frequencies we extract from our signal. i.e. the basis functions. |

Let's delve into this a bit deeper, but first if you are uncertain about normalised frequency, Ω, you should revise this now.

Consider the complex exponential term in the DTFT in comparison with the DTF (remember the exponential describes the frequency components we are searching for in our time series):

$$e^{-jn\ \Omega} \qquad \& \qquad e^{-jn\ 2\pi k/N}$$

I've rearranged the order of the terms to make things more visible.

The first point to note is that by convention the DFT is described in terms of frequency in Hertz (rather than rad/s) hence the presence of the '2π' term.

We can now see that: $\Omega = 2\pi k/N$

Unravelling this further we know that Ω is 'radians per sample period', hence *k/N* is 'cycles per sample period' and we have said that *k* is an integer ranging $0 \le k \le N-1$, thus our lowest frequency is zero, and our highest frequency is *N-1/N* cycles per sample – just below the maximum frequency we can uniquely sample. If we have an 'N' sample long discrete sequence then we can resolve frequency into 'N' frequency steps.

The key take-home here is that we have specified the frequencies we are looking for in relation to the length and sampling period of the discrete time domain signal – we shall see this makes a lot of sense!

**Example:**

If we have a discrete time domain signal of 10 samples (N = 10) and the sample rate is 0.1 second.

K will take integer values over the range $0 \le k \le N-1$, so $0 \le k \le 9$

k/N will range: $0 \le (k/N) \le N-1/N$, or $0 \le (k/10) \le 0.9$

k/N is still normalised, in the form of 'cycles per sample period' so we need to re-dimentionalise it by taking sample period into account. Recall:

1) $\Omega = 2\pi k/N$ (defined in this section) and 2) $\Omega = \omega T = 2\pi f T$ (from definition of Ω)

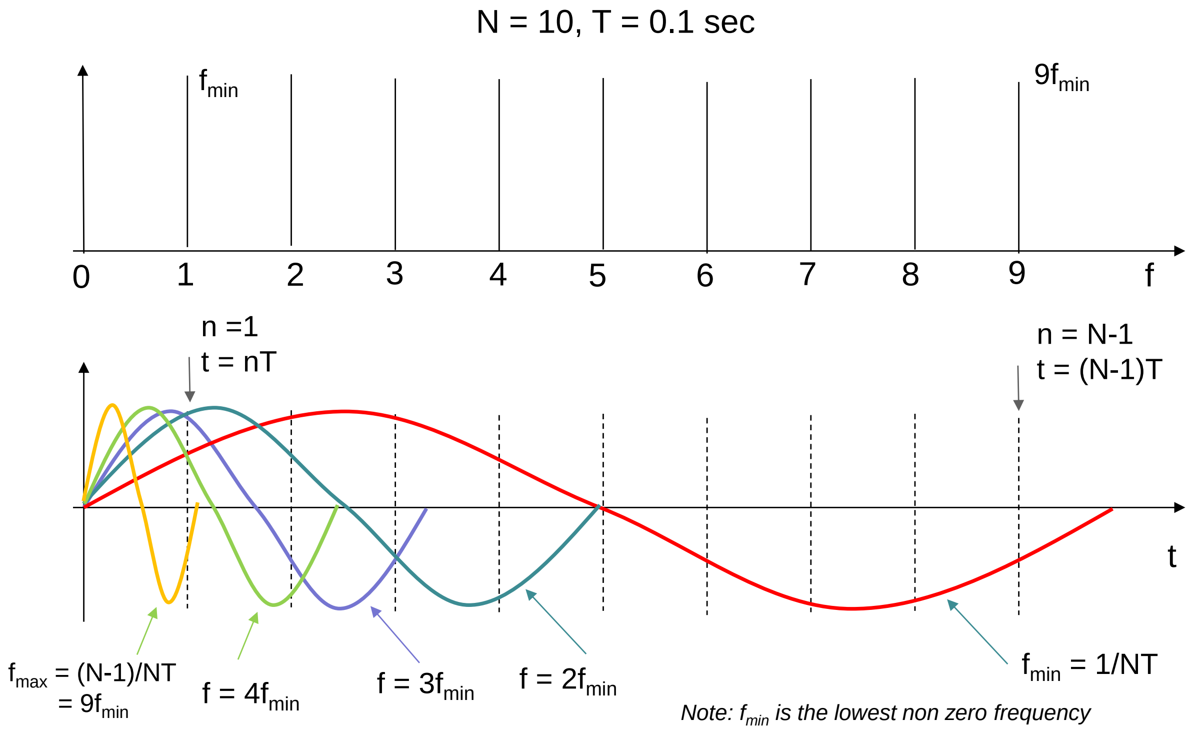
Equating 1 & 2:

$f = k/NT$

Hence f will range: $0 \le k/NT \le (N-1)/NT$ or $0 \le 1/(10*0.1) < 9/(10*0.1)$

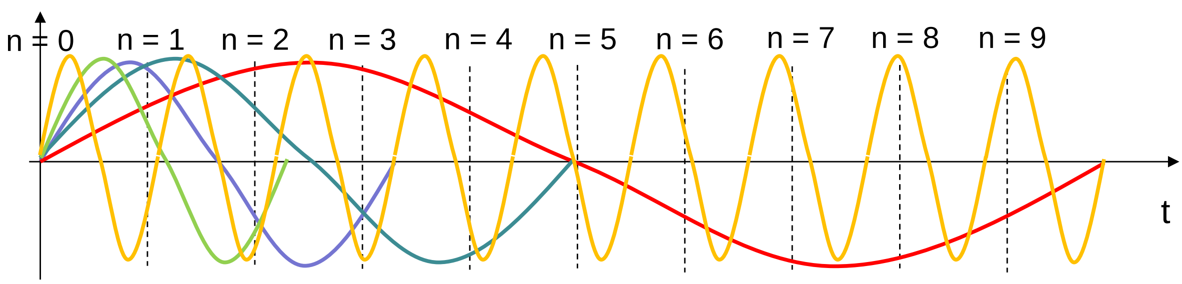
i.e. 0 – 9 Hz, in steps of 1 Hz.

We can look at the frequency components defined by the DFT diagrammatically in both domains:




### 1.6.5 The Fast Fourier Transform FFT

The DFT is computationally very expensive to perform – but let us take a look at the lowest and highest frequency components again:




Consider the low frequency component (Red): *There are many more samples than we need to identify this harmonic, so we could reduce the number of samples for which we calculate the transform.*

Consider the high frequency component (Yellow): *There are many more cycles than we need to identify this harmonic. We could reduce the period of time over which we calculate the transform*

The ***Fast Fourier Transform*** describes a range of algorithms that increase the speed of the calculating a DFT. For long sequences a DFT is very computationally expensive as the number of computations follows $N^2$

By recognising that in any 'N' long sequence, there are many more cycles than needed to find the high frequencies, and many more samples than required to calculate the low frequencies, the computational complexity can be reduced to *NlogN*

For large N this can be many orders of magnitude.

### 1.6.6 Frequency spectrums

So far, we have tended to think about spectrums made up of individual frequency components – each one represented by an impulse in the frequency domain:

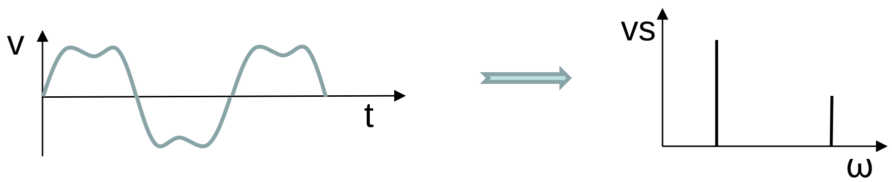



***Fig 2. A periodic time domain waveform maps to individual harmonic components in the frequency domain.***

We know that the frequency domain y axis units are *signal_units*seconds* as a consequence of the correlation integral transform – hence if the time sequence is infinite duration the frequency components will have a true impulse function in the frequency domain with infinite magnitude; practically we will have time signals of finite duration and we might divide the frequency domain magnitudes by the time signal length to recover the information we want (the information we normally want is the magnitude of the time domain phasors).

But how about signals which are continuous in the frequency domain? for example:

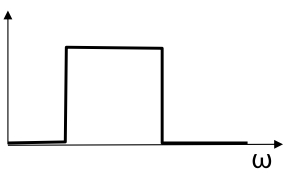



***Fig 3. This signal is in the frequency domain – what would it look like in the time domain?***

We will explore this in a few ways, but to start we can start to think of this by looking at the math and considering the answer to the question posed right at the end of section 1.4.4 regarding the Fourier transform of an impulse.

Time domain

Frequency domain

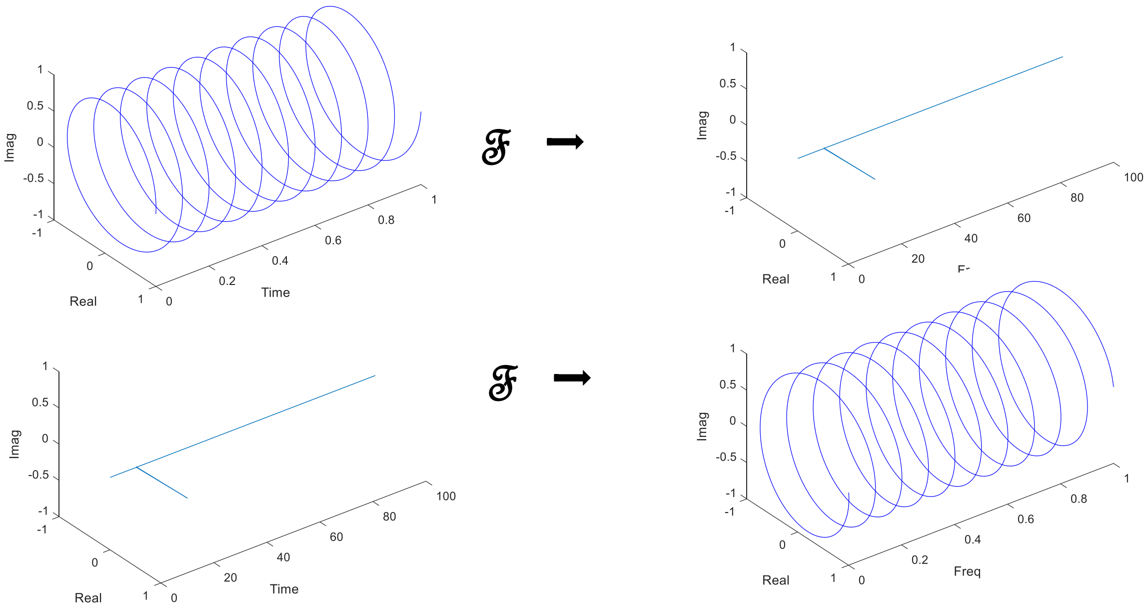



***Fig 4. This Fourier transform and its inverse are pretty much identical- only the sign of the exponent changes, hence it should be no surprise that an impulse in the time domain looks like a helix in the frequency domain***

A helix in the time domain (which we have up to now thought of as our basis functions in the time domain) produces an impulse in the frequency domain; but also the inverse Fourier transform of a helix in the frequency domain produces an impulse in the time domain!

Those of you who are following this from the pure math perspective are likely quite happy with this, those who have been using physical interpretation might need to spend a little time, after all, what does a helix in the frequency domain represent physically?

Consider first our time domain phasor (helix) and its Fourier transform

- Increasing or shortening the duration of our time domain helix causes the height of the impulse in the frequency domain to grow or shrink.
- Stretching or compressing the helix causes the impulse to shift along the frequency axis.
- Shifting the time domain helix to the left or right causes the frequency domain impulse to rotate around the frequency axis

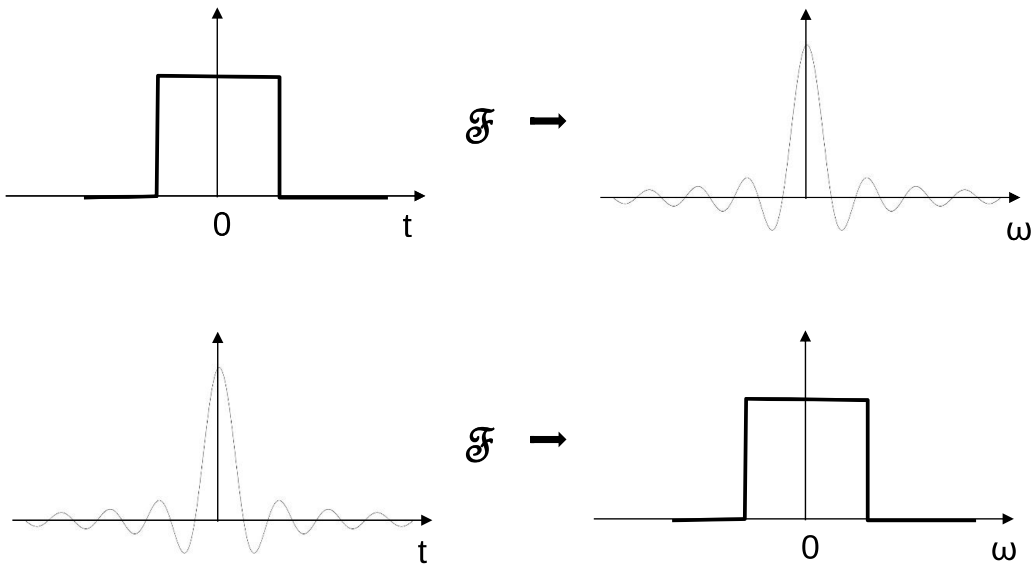
- Having a real-valued representation of the helix (i.e. sine or cosine) produces a second mirror-image negative frequency impulse (backwards and forwards rotating frequency components).

Now think about this from the case of the time domain impulse and its Fourier transform of a helix. What happens to the time domain impulse and the physical significance if we: 1) increase or shorten the length of the frequency domain helix?; 2) stretch or compress the frequency-domain helix?; 3) shift the helix left or right?; 4) have a real-only valued frequency domain signal?

If you can work through how the changes in one domain affect the representation of the signal in the other domain then you will start to have a good understanding; the mapping between the domains is revealed.

We have also illustrated a very useful practical fact. If there is only one thing you remember from this lecture, it should be that high rates of change in the time domain (d/dt) implies high frequency components are present in a signal.

But what about the rectangular spectrum we introduced earlier? Some of you may know the standard result that the Fourier transform of a rectangular function is the sinc function (i.e. of the form sin(x)/x), thus we can (correctly) suppose that the inverse transform of a rectangular frequency function is a time domain sinc function:




We have previously described that the reason we get negative frequency (and/or time components) is that we are dealing with real-valued signals, rather than analytic representations (this might also explain to you why in the past you will have often considered functions/signals reflected about x = 0).

Back in the physical world, there are two interesting things we can pull out of this:

1) If you just looked at zero crossings, the sinc function would at first sight appear to only be comprised of a single frequency (as zero crossings occur with the same period); Indeed the expression for a sinc function is only appears to have a single frequency (sin(x)) term. The interesting thing here is that modulating the magnitude of our basis functions 'smears' their energy over a range of frequencies – if you have a sine wave that is growing or shrinking the energy is no longer contained at a single frequency.
2) The sinc function is infinite in duration – hence it can be comprised of an infinite number of frequency-domain impulses.

### 1.6.7 Noise signals

Another way to think of continuous frequency spectrums is to consider 'noise' - the most commonly encountered random signal.

Noise often describes unwanted frequency components in the background of a signal. Many processes that create noise are unknown (if they are known it is common to refer to them as 'interference') and they are typically described using stochastic tools: probability theory etc. We can intuitively see that noise will contain a wide range of frequencies.

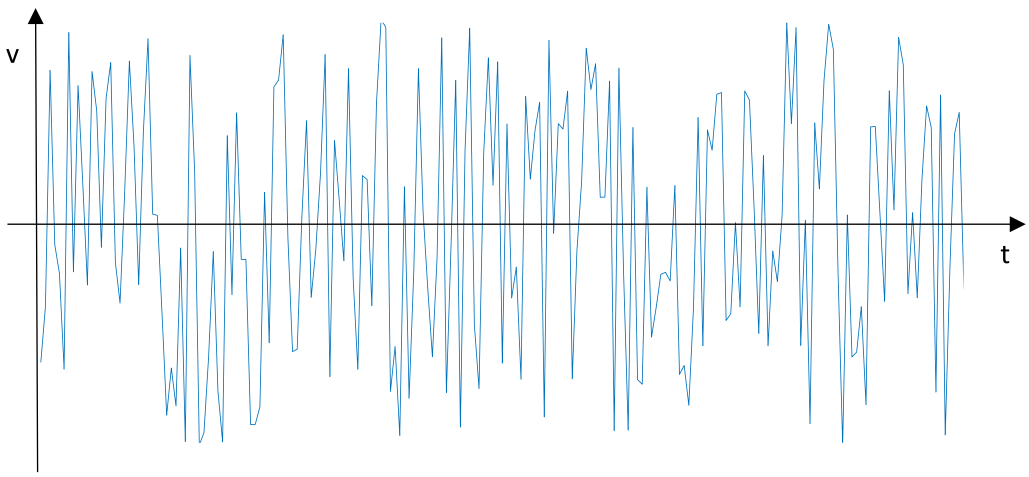



***Fig 4. Noise is a random variation***

In the time domain it doesn't make sense to describe the instantaneous value of such a signal at a point in time – we use probability density functions to tell us the range of values the signal will take over a period of time, and variance to capture the average magnitude.

### 1.6.8 Power spectral density

The PSD or Power Spectral Density is an application of the frequency domain that you will come across often if you deal with practical signals as it is a commonly used way to describe the frequency spectrum of a signal where the signal has noise or broadband components

In the last section we discussed that a probability distribution function can be used to describe a random signal but it cannot directly tell us about the frequency spectrum of a signal – note there are some special cases, such as Gaussian White Noise – where we describe a signal that has a Normal distribution and a flat frequency spectrum. However, we can say there must be some physical equivalence between a signal whether it is represented in the time domain or the frequency domain, and we draw this equivalence through power (or energy).

The energy must be the same whether the signal is represented in the time or frequency domains so we can say:

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega$$

This is known as **Parseval's theorem** and it says that if we sum the energy of the signal over all time, it will equal the sum of the energy over all frequencies. We can also apply this over a particular period of time to capture the power.

The Power Spectral Density (PSD) can be derived by substituting the expression for the Fourier transform into Parseval's Theorem (we aren't going to go through the maths in this unit). The PSD expresses the magnitude of signal power within a particular bandwidth. Often the signal is a field (root power) quantity, and we square it to give a value proportional to power, and the units of the PSD are then Signal^2 per Hertz bandwidth.

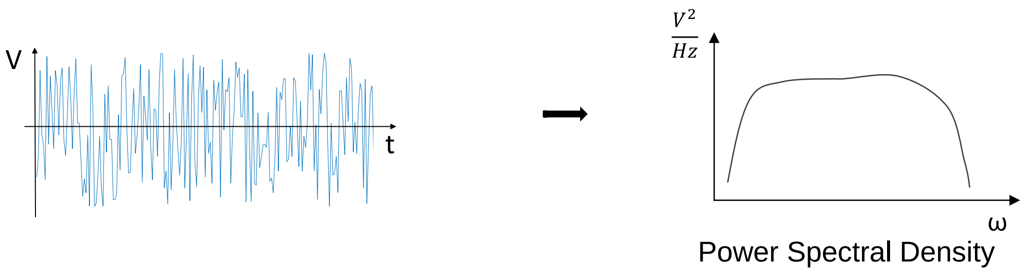



***Fig 4. Noise spectrum***

Although Paseval's theorem provides useful 'constraints' when trying to consider frequency spectrums, it does not in itself imply a continuous spectrum. A smooth continuous spectrum only appears from a random signal if the signal continues for all time – although you can begin to approximate this for long signals - instead it tells us where the frequency range over which a signals energy can lie. For a short random signal, the frequency spectrum will be random too!

The PSD finds real utility when we have discrete domains in the frequency range – for example when we use the DFT or FFT.  This is because the times series is finite and as a result the DFT captures frequencies a little either side of the central frequency of the correlation and we think of this as creating a range of ‘frequency bins’.  This has the effect of making the magnitude in each bin a function of the bin width – which is in turn determined by sampling period and sequence length. Hence, when taking the FFT of a random signal the sampling period will affect the result.  Using the PSD prevents this because in automatically normalises to the bin width.

Just to confuse things, a single frequency component is not affected by bin width in the same way.

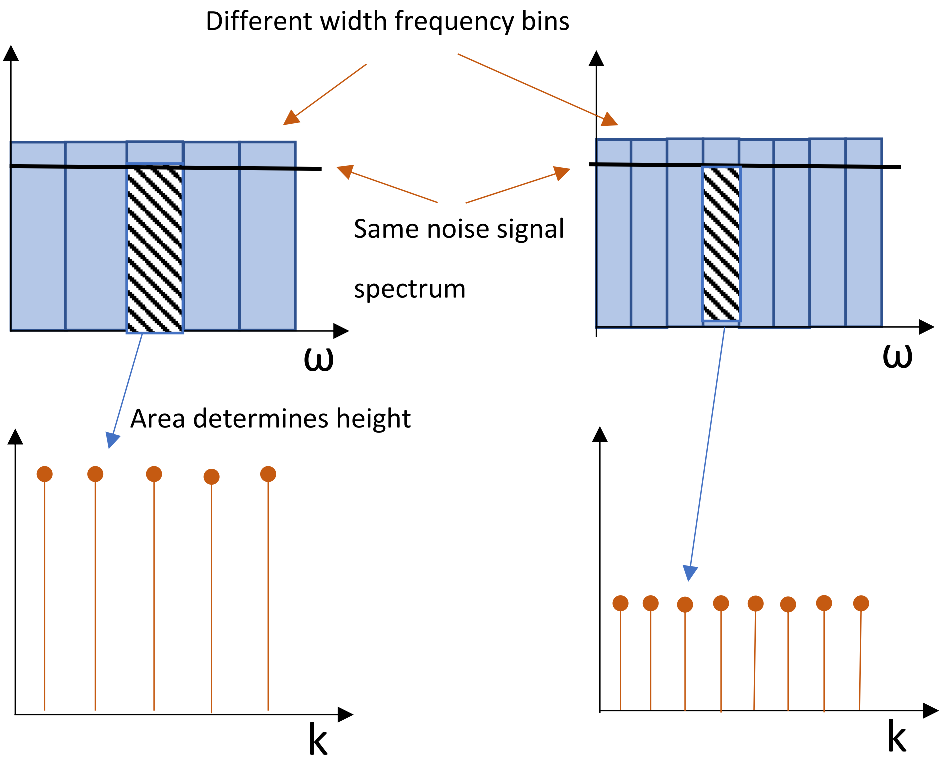



***Fig 5. Illustration of how the bin width determines how much energy is captured from the same noise signal***

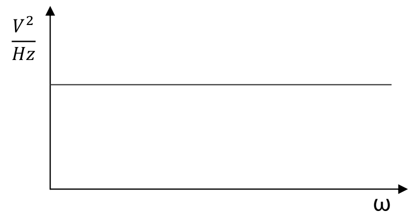
You can find a video explaining the effect here:

[What is a Power Spectral Density (PSD)?]

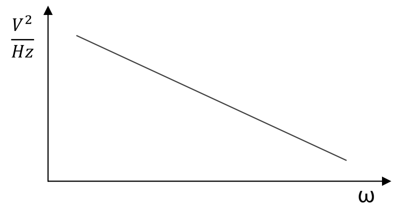
They call the FFT ‘autopower’, and bin width is referred to as ‘spectral lines’

### 1.6.9 Colours of Noise - Noise signal spectrums

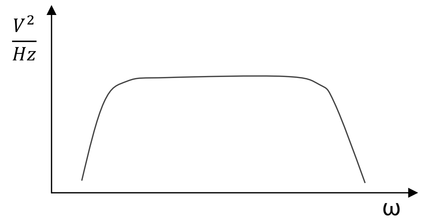
Noise is commonly classified according to the shape of its frequency spectrum. Some of the colours (white for example) might be familiar to you.




*White noise has a flat spectrum*




*Pink noise has a 1/ω spectrum*




*Band limited white noise is constant over some range*

### 1.6.10 Test yourself

1) What does discretisation in the frequency imply?
2) Why would we want to use a discrete frequency domain?
3) What do the acronyms DTFT, DFT and FFT stand for?
4) How does the length of the sampled signal in the time domain affect the frequencies of the DFT?
5) What do the range of algorithms described as 'FFT' achieve compared to the DFT?
6) If a helix in the time domain transforms to an impulse in the frequency domain, what is the inverse transform of a helix in the frequency domain?
7) Why does a high rate of change in the time domain imply high frequency components in the signal?
8) What is a PSD function?
9) How does Parseval's Theorem link a random time domain signal with a PSD?
10) What are: white noise?; Pink noise?; Band-limited noise?