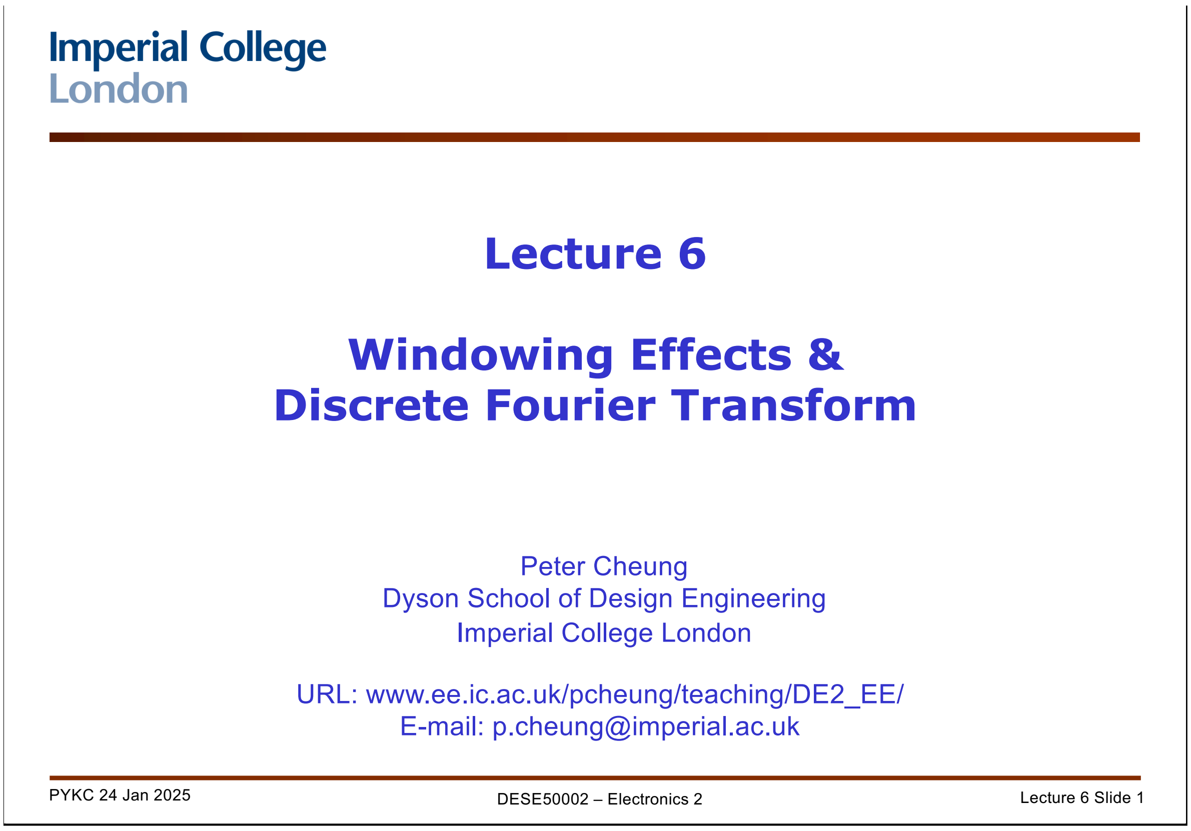


Imperial College London

Lecture 6

Windowing Effects & Discrete Fourier Transform

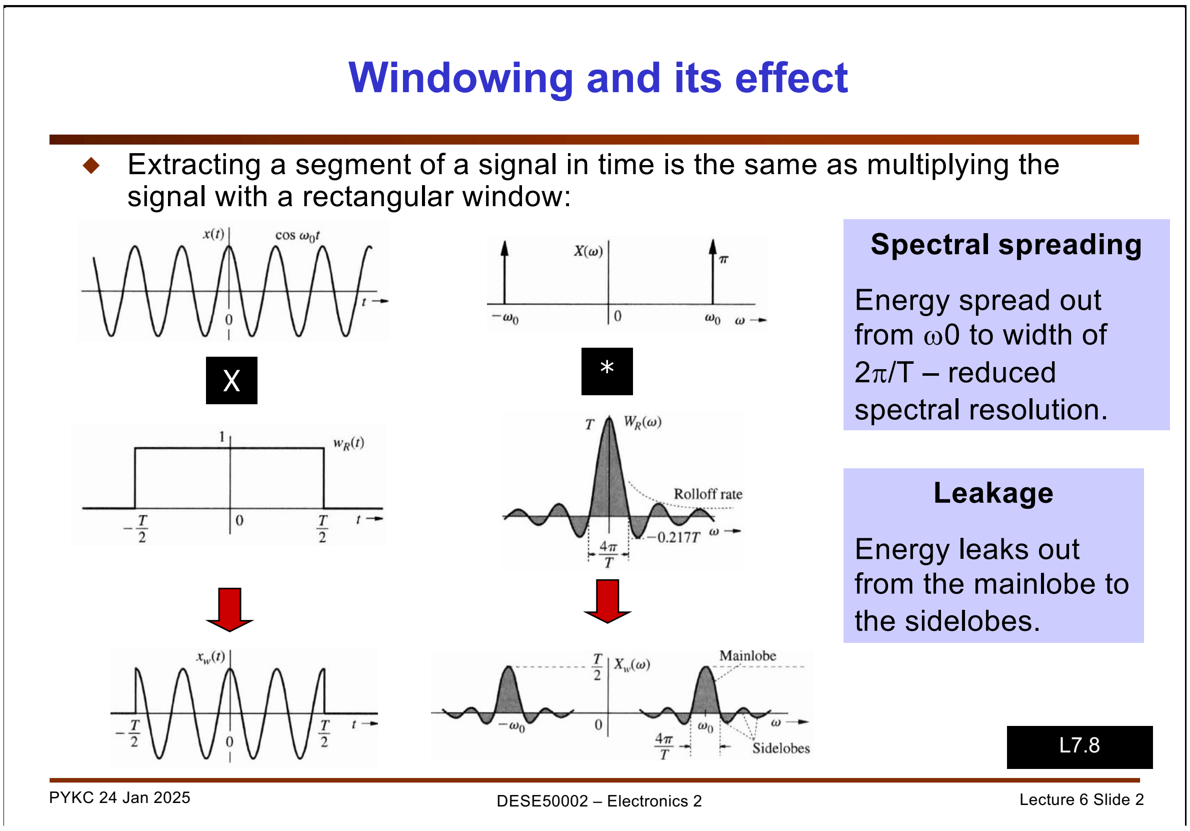
Peter Cheung
Dyson School of Design Engineering
Imperial College London

URL: www.ee.ic.ac.uk/pcheung/teaching/DE2_EE/
E-mail: p.cheung@imperial.ac.uk

In this lecture, I will be examining the impact of extracting a portion of a signal and find the spectrum of this extracted portion instead of the signal. This process of taking a portion of signal is known as "windowing".

Then I will discuss the calculation of the Fourier transform of a signal on a computer using discrete method, known as Discrete Fourier transform or DFT.

Finally I will explain how to calculate energy of a signal in the frequency, instead of the time, domain.



Windowing and its effect

- Extracting a segment of a signal in time is the same as multiplying the signal with a rectangular window:

$x(t)$
$\cos \omega_0 t$
t
0
$X(\omega)$
π
$-\omega_0$
0
ω_0
ω
X
*
1
$w_R(t)$
$-\frac{T}{2}$
0
$\frac{T}{2}$
t
T
$W_R(\omega)$
Rolloff rate
$-0.217T$
ω
$\frac{4\pi}{T}$
$x_w(t)$
$-\frac{T}{2}$
0
$\frac{T}{2}$
t
$\frac{T}{2}$
$X_w(\omega)$
Mainlobe
$-\omega_0$
0
$\frac{4\pi}{T}$
ω_0
ω
Sidelobes

Spectral spreading

Energy spread out from ω0 to width of $2\pi/T$ – reduced spectral resolution.

Leakage

Energy leaks out from the mainlobe to the sidelobes.

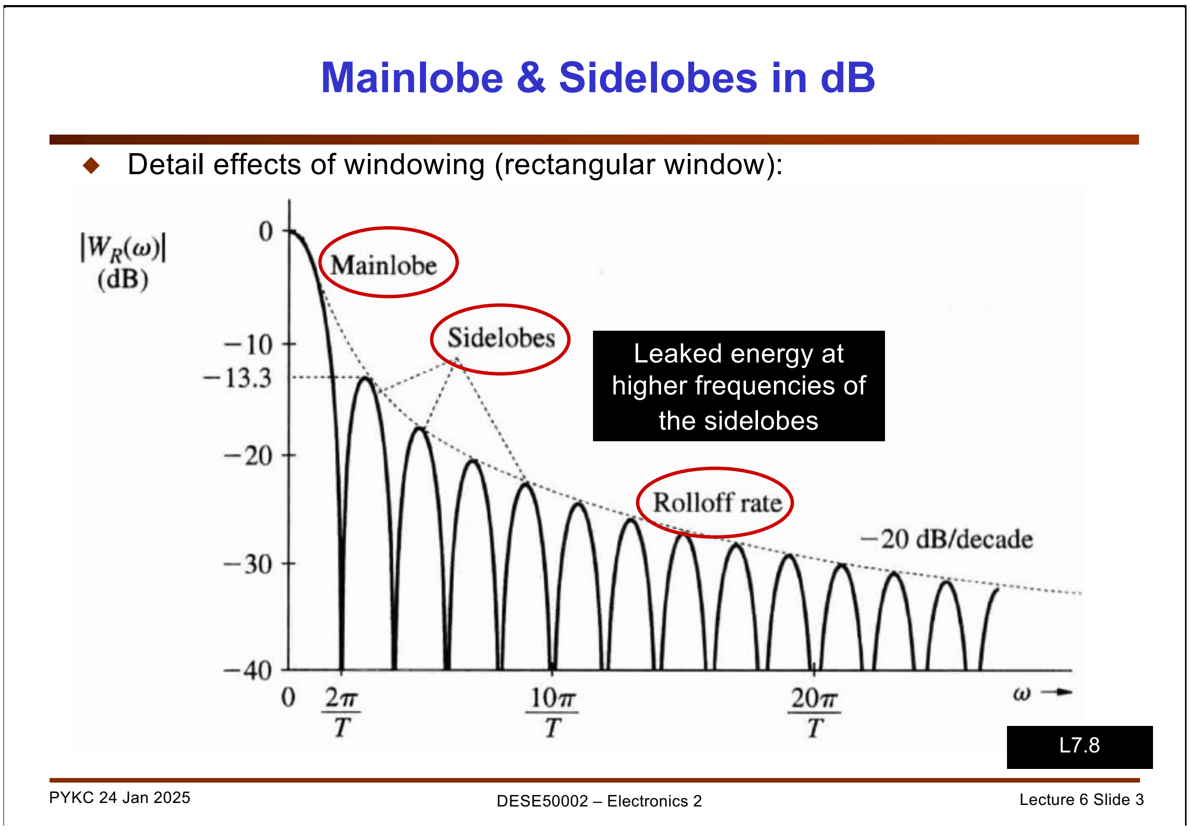
L7.8

Extracting a portion of signal from an everlasting sinusoidal signal is the same as multiplying the everlasting sinewave with a rectangular function as shown.

As mentioned in the last lecture, multiplying in the time domain is equivalent to an operation known as “convolution” in the frequency domain. We will not examine convolution until a later lecture.

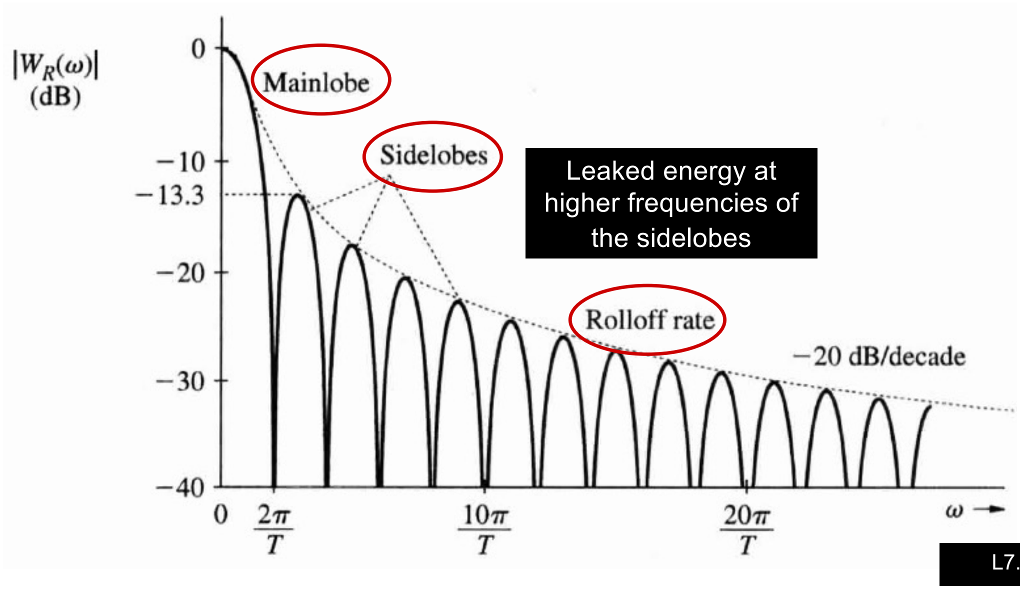
It is sufficient for you to know for now that convolution results in the spectrum of the original sinewave, which is two spikes (impulses) at $\pm\omega_0$, being modified by the spectrum of the rectangular function as shown above.

The impact of applying the rectangular window (i.e. function) in the time domain is to spread out the energy of the sinusoid around the impulses.



Mainlobe & Sidelobes in dB

- Detail effects of windowing (rectangular window):

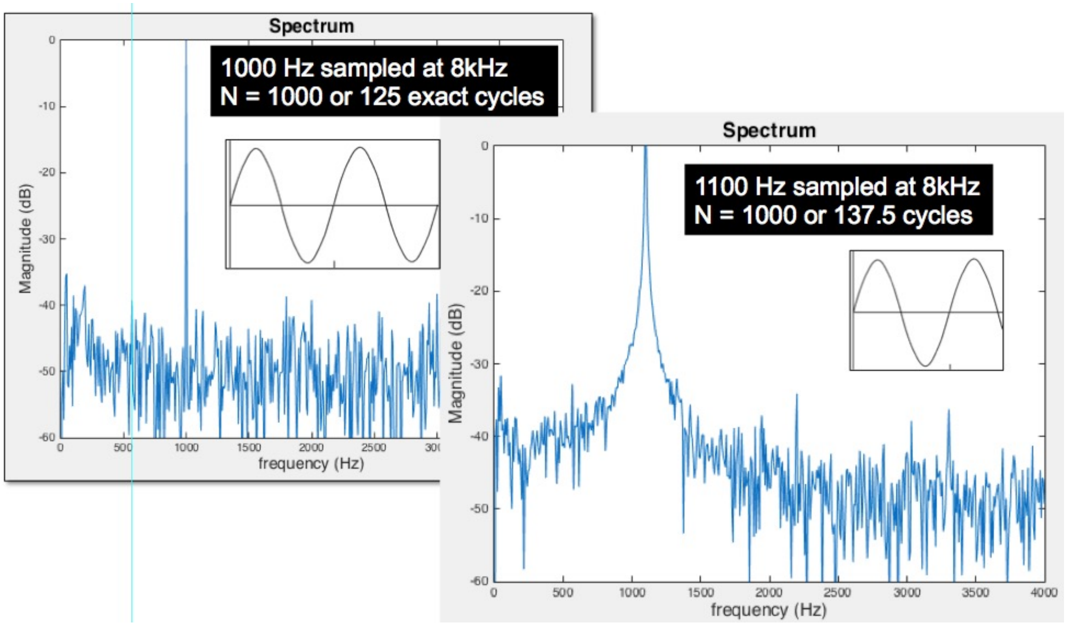


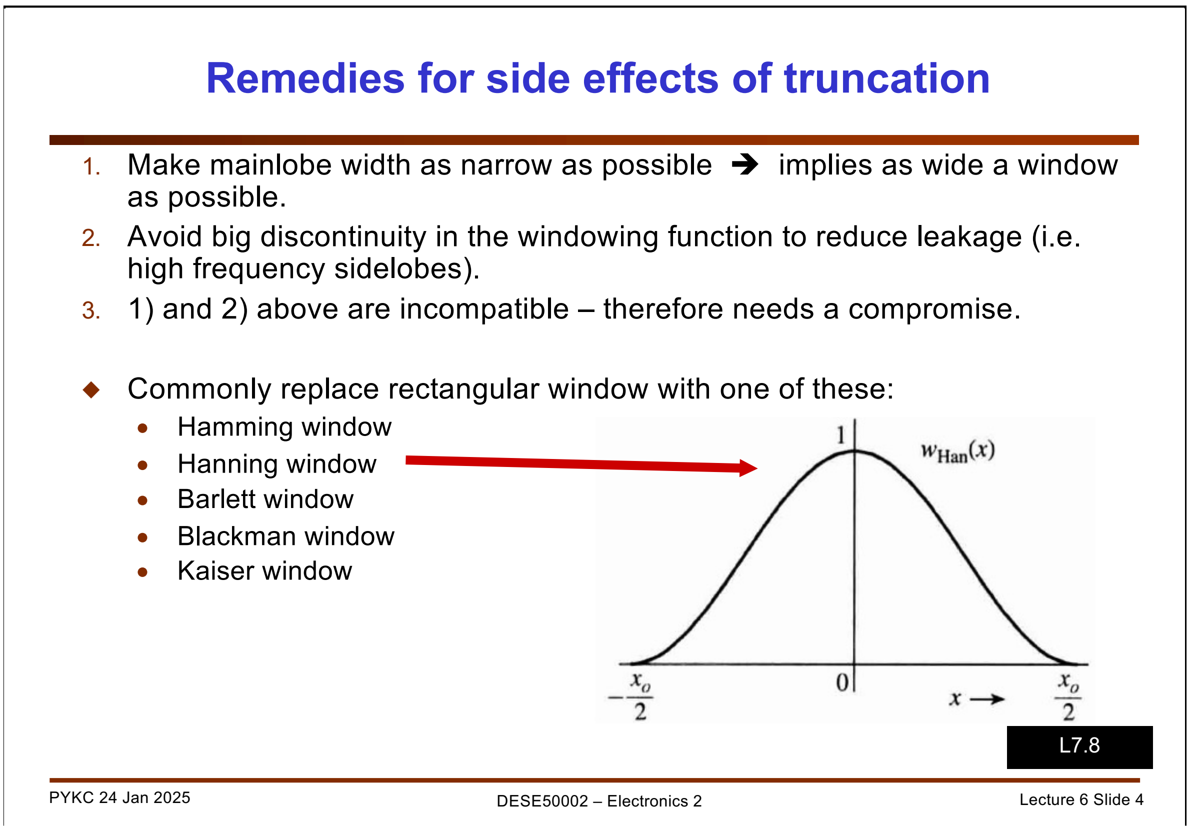
Leaked energy at higher frequencies of the sidelobes

L7.8

Let us examine the magnitude spectrum of the rectangular function (in dB). It has a main region where most of the energy lies. This is call the mainlobe of the spectrum. In addition, energy is also spread to its neighbouring lobes, the sidelobes. This is known as "leakages" – energy leaked from the main frequency component.

The reason for such energy spreading to higher frequency is that the rectangular window causes discontinuity (or abrupt changes) at the edges of the window. For example, in Lab 2 exercise 3, you have discovered the difference between the spectrum for a 1000Hz and a 1100Hz sinewave with 1000 samples extracted and sampled to 8000Hz. The 1000Hz has 125 exact cycles in the rectangular window. However the 1100Hz sinewave has 137.5 cycles extracted by the rectangular window.

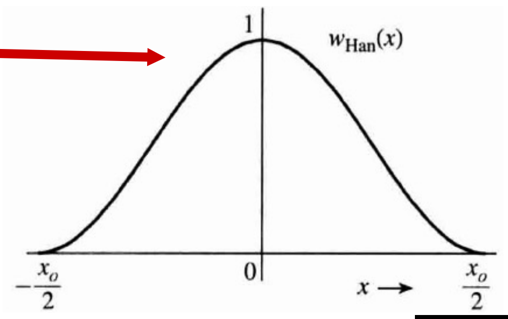




Remedies for side effects of truncation

1. Make mainlobe width as narrow as possible ➔ implies as wide a window as possible.
2. Avoid big discontinuity in the windowing function to reduce leakage (i.e. high frequency sidelobes).
3. 1) and 2) above are incompatible – therefore needs a compromise.

◆ Commonly replace rectangular window with one of these:
   - Hamming window
   - Hanning window
   - Barlett window
   - Blackman window
   - Kaiser window



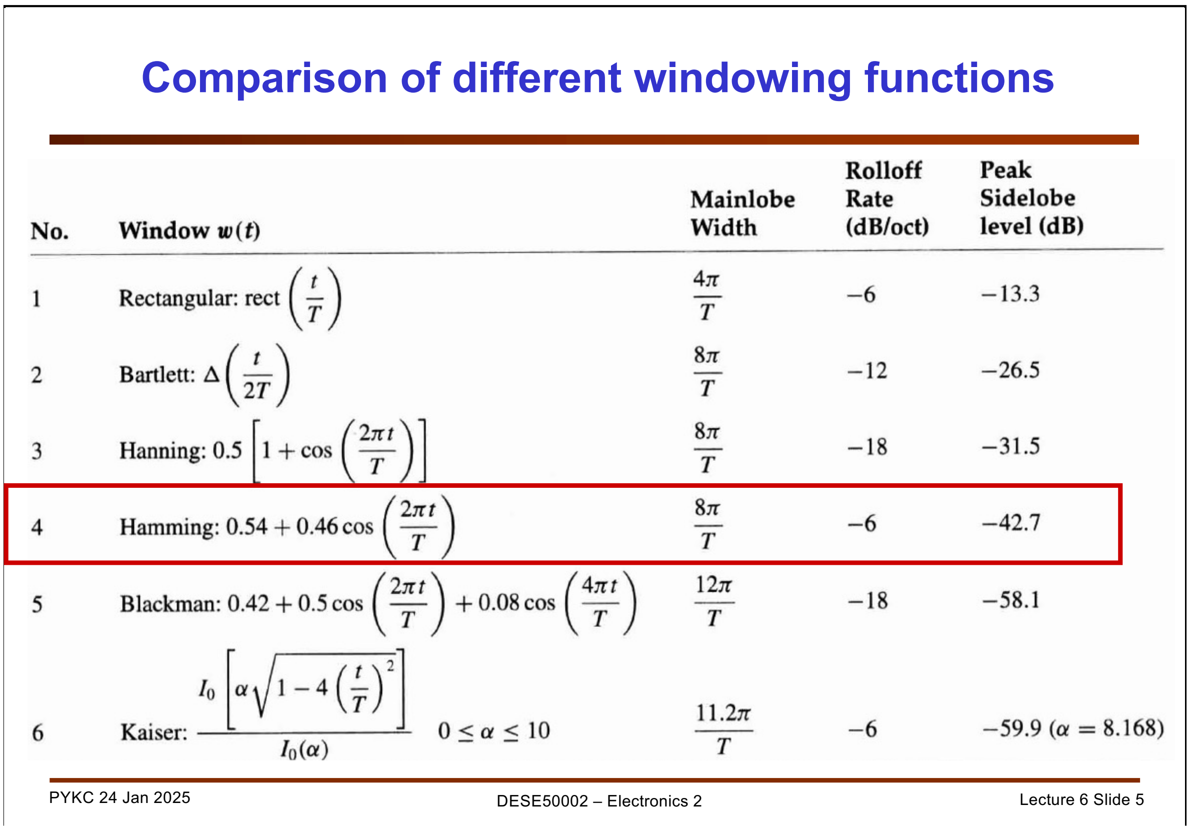
L7.8

Instead of using a rectangular window function to "extract" portion of the signal to analyse, it is far better to apply a smooth window function, where the edges of the window tails off gradually. In the lab we, use a Hamming window. Shown here is the plot of the Hanning window, which is really very similar.

There are two impact on applying such a window:

1.It reduces the sidelobes substantially and therefore reduces the leakages.

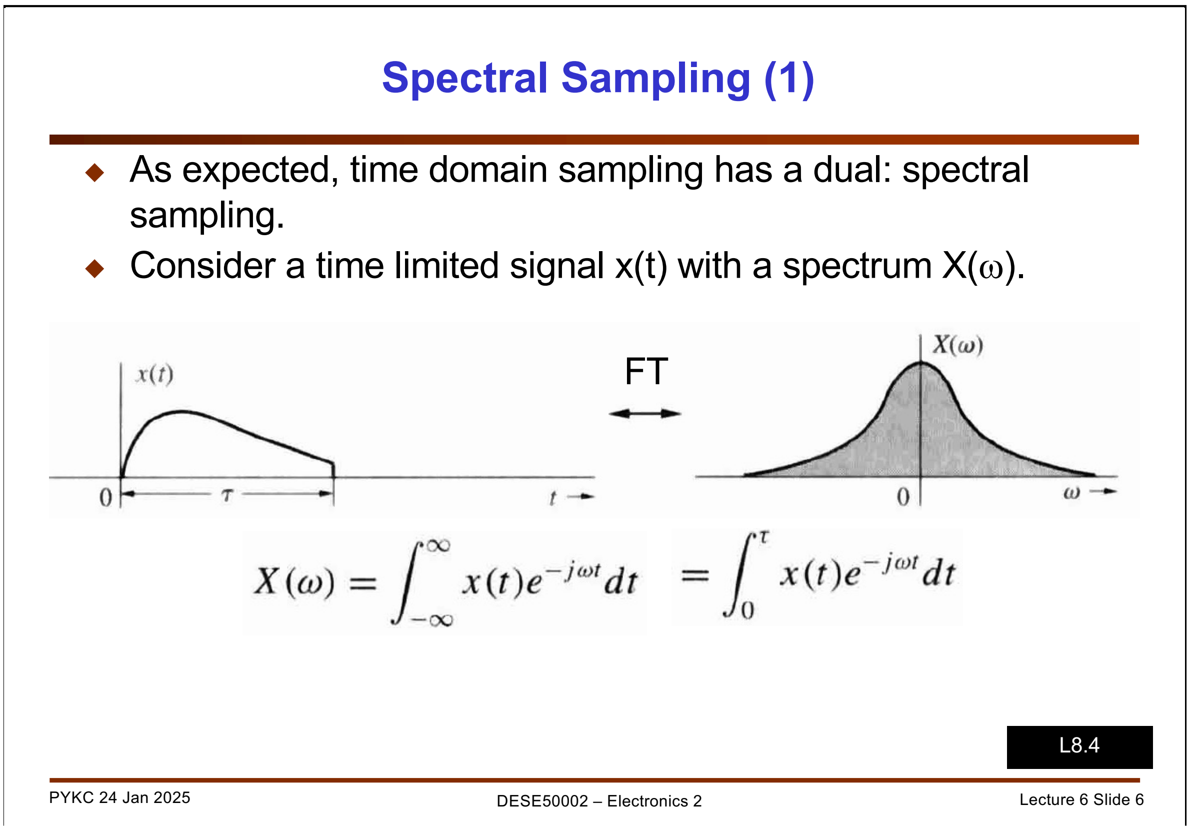
2.The total energy is reduced, but energy of the signal at each frequency relative to each other is unaffected. In other words, if your unwindowed signal is the sum of two sinusoids ω1 and ω2, the magnitudes of both are reduced by using one of these windows, but their ratio remains the same.



Comparison of different windowing functions

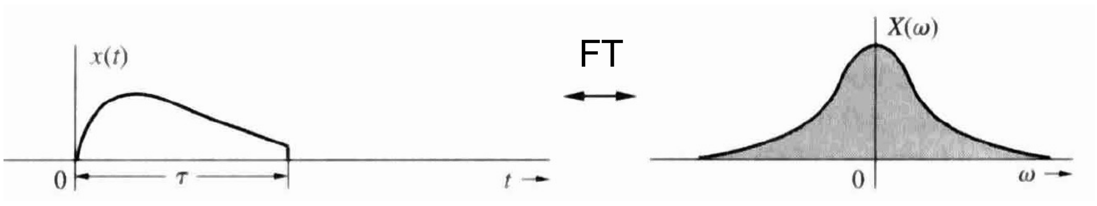
No.	Window $w(t)$	Mainlobe Width	Rolloff Rate (dB/oct)	Peak Sidelobe level (dB)
1	Rectangular: $\text{rect}\left(\frac{t}{T}\right)$	$\frac{4\pi}{T}$	-6	-13.3
2	Bartlett: $\Delta\left(\frac{t}{2T}\right)$	$\frac{8\pi}{T}$	-12	-26.5
3	Hanning: $0.5\left[1+\cos\left(\frac{2\pi t}{T}\right)\right]$	$\frac{8\pi}{T}$	-18	-31.5
4	Hamming: $0.54+0.46\cos\left(\frac{2\pi t}{T}\right)$	$\frac{8\pi}{T}$	-6	-42.7
5	Blackman: $0.42+0.5\cos\left(\frac{2\pi t}{T}\right)+0.08\cos\left(\frac{4\pi t}{T}\right)$	$\frac{12\pi}{T}$	-18	-58.1
6	Kaiser: $\dfrac{I_0\left[\alpha\sqrt{1-4\left(\frac{t}{T}\right)^2}\right]}{I_0(\alpha)}$ $\quad 0 \le \alpha \le 10$	$\frac{11.2\pi}{T}$	-6	-59.9 ($\alpha = 8.168$)

Here is a table of the various windows commonly used as a function of time, the width of the mainlabe (the narrower, the better), the rate at which the sidelobe energy falls away (the faster rate the better), and the height of the first sidelobe.



Spectral Sampling (1)

- As expected, time domain sampling has a dual: spectral sampling.
- Consider a time limited signal x(t) with a spectrum X(ω).

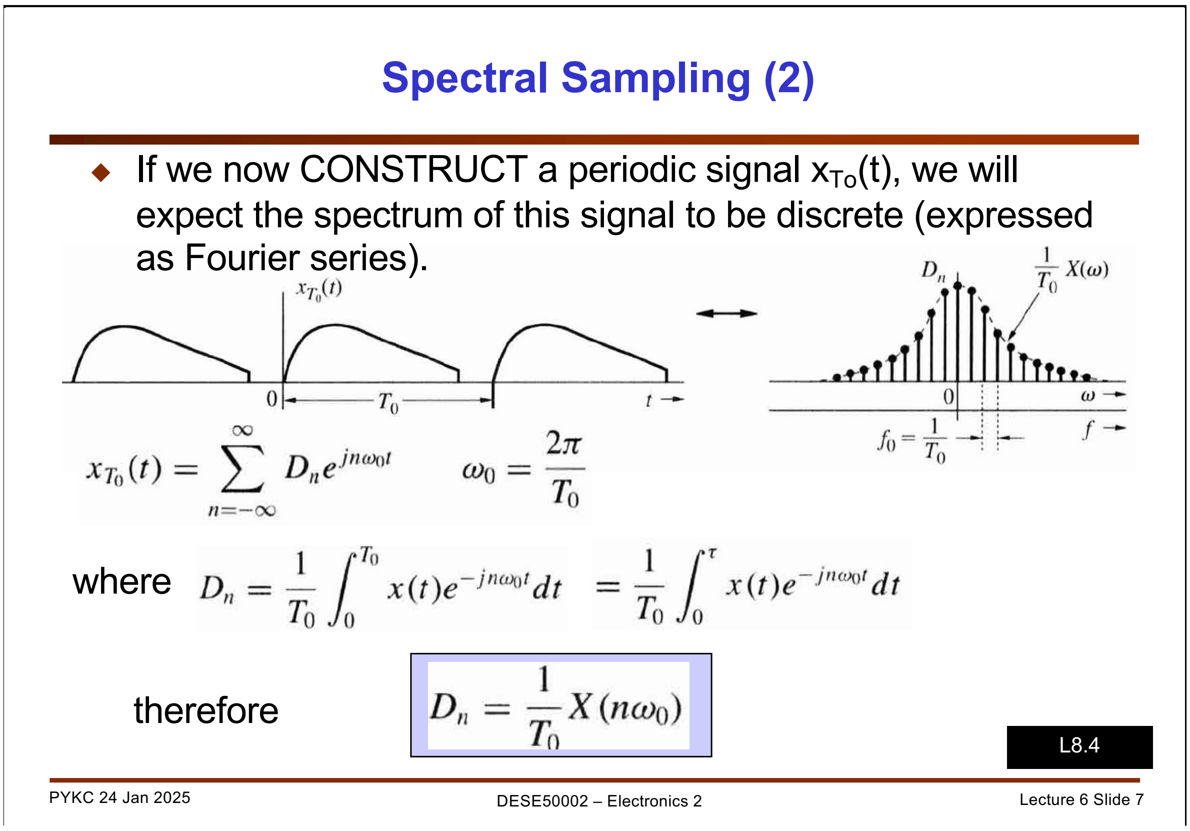


$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \quad = \int_{0}^{\tau} x(t)e^{-j\omega t}dt$$

L8.4

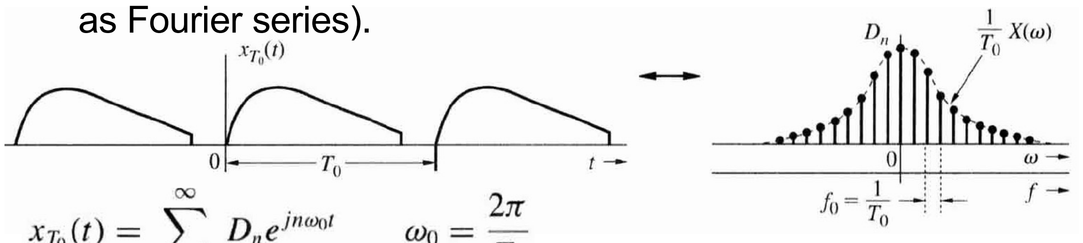
Next we will examine how the computer calculate the spectrum – using discrete values instead of continuous values.

First remind yourself the definition of Fourier Transform.



Spectral Sampling (2)

- If we now CONSTRUCT a periodic signal $x_{T_0}(t)$, we will expect the spectrum of this signal to be discrete (expressed as Fourier series).



$$x_{T_0}(t) = \sum_{n=-\infty}^{\infty} D_n e^{jn\omega_0 t} \qquad \omega_0 = \frac{2\pi}{T_0}$$

where $$D_n = \frac{1}{T_0}\int_0^{T_0} x(t)e^{-jn\omega_0 t}dt = \frac{1}{T_0}\int_0^{\tau} x(t)e^{-jn\omega_0 t}dt$$

therefore $$\boxed{D_n = \frac{1}{T_0}X(n\omega_0)}$$

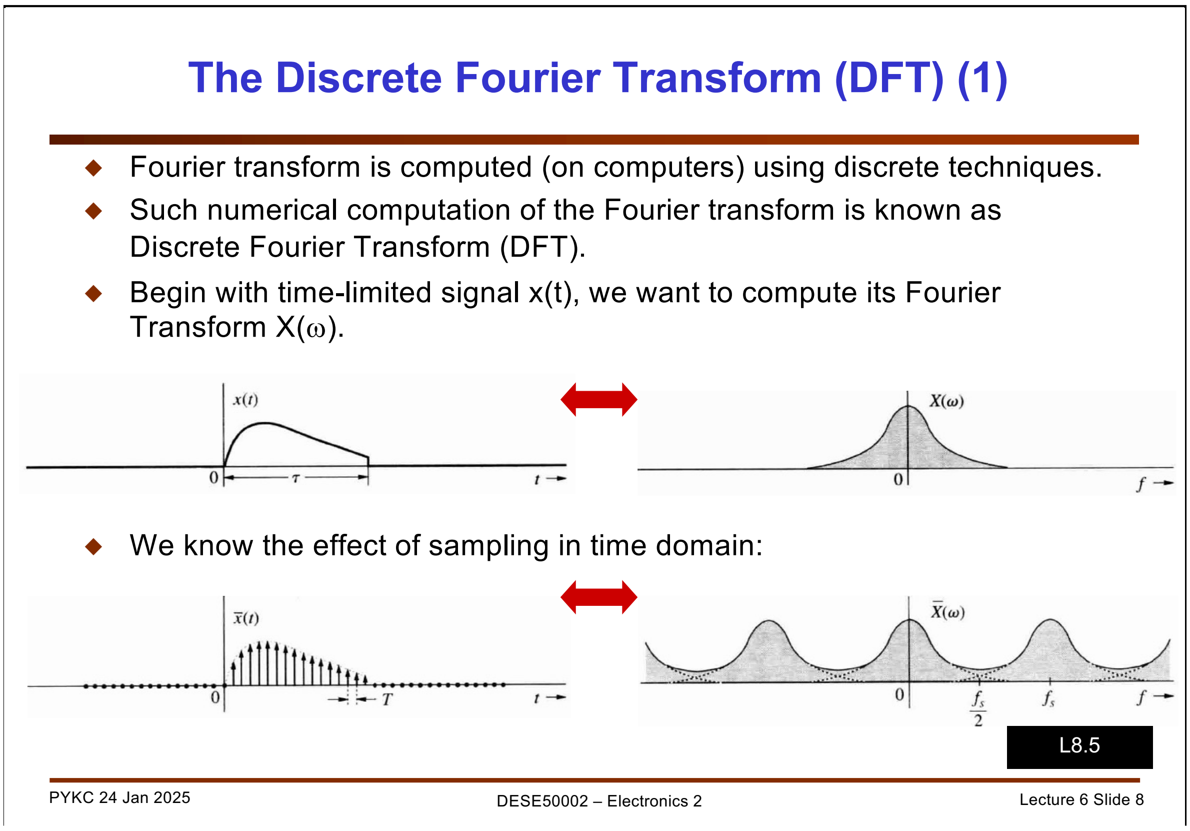
L8.4

When you use the Matlab function fft(sig) to compute the spectral component values, you perform the Discrete Fourier Transform (DFT) calculation using a fast algorithm. The fast algorithm is known as Fast Fourier Transform. How this is done is not important. It is sufficient to know that a straight calculation of DFT on a N sample signal has a computation complexity of order(N^2). Using FFT, we reduce this order(N log N). Say, if N 1000, using FFT instead of DFT would be over 300 times faster!

To compute the FFT or DFT of a time domain signal, we first extract the signal with an appropriate window (N samples in a window of T_0), then make up a periodic signal as shown. (This is the formulation used inside the computer algorithm. The signal does not really exists as everlasting periodic signal does not happen in real life!)

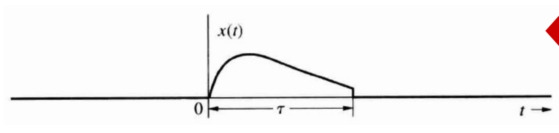
The DFT/FFT algorithm basically compute the Fourier coefficients of this repetitive signal using the standard Fourier series equation as shown.

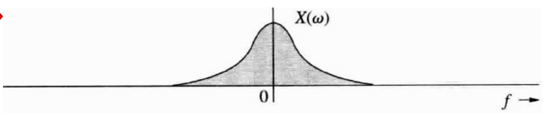
The maths here just shows that the effect of DFT is to take discrete samples of the Fourier Transform of our windowed signal at frequency steps of $f_0 = 1/T_0$.



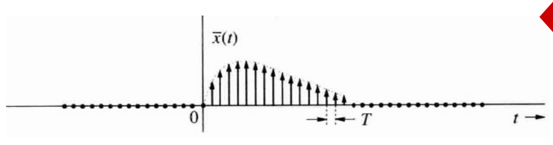
The Discrete Fourier Transform (DFT) (1)

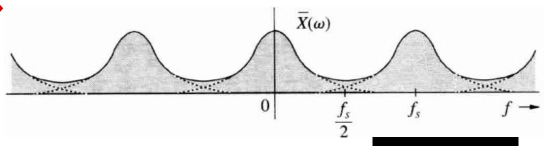
- Fourier transform is computed (on computers) using discrete techniques.
- Such numerical computation of the Fourier transform is known as Discrete Fourier Transform (DFT).
- Begin with time-limited signal x(t), we want to compute its Fourier Transform X(ω).





- We know the effect of sampling in time domain:





L8.5

Let us examine DFT from a conceptual, instead of mathematical, point of view.

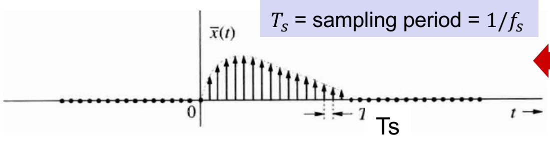
When we take a continuous time signal x(t) and obtain its spectrum using Fourier transform, we get X(ω).

If we sample the continuous time signal, the resulting discrete time signal has a different spectrum as compared to that of the original signal. In fact, the original spectrum is repeated indefinitely at each frequency location ±n fs, where fs is the sampling frequency and n is all integers (plus or minus) n = ±1, ±2 ……

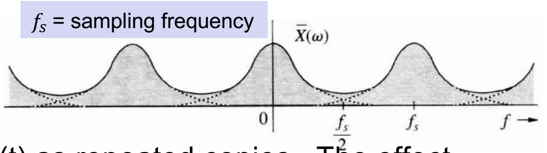
This result is very important. It says that the sampling process modifies the spectrum of the signal in this particular way that has implications to how often we need to sample (i.e. what fs to choose), how we can avoid corrupt the signal through the sampling process, and finally, how can we get back the original continuous time signal from the discrete time signal.

The Discrete Fourier Transform (DFT) (2)

T_s = sampling period = $1/f_s$

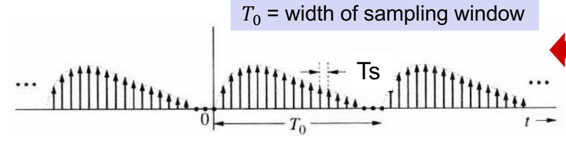


f_s = sampling frequency

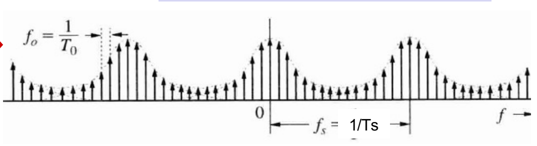


- Now construct the sampled version of x(t) as repeated copies. The effect (from slides 6-8) is sampling the spectrum.

T_0 = width of sampling window



f_0 = frequency increment = $1/T_0$



Number of time samples in T_0

$$N_0 = \frac{T_0}{T_s}$$

Number of frequency samples in f_s

$$N'_0 = \frac{f_s}{f_0}$$

$$N_0 = \frac{T_0}{T_s} = \frac{1/f_0}{1/f_s} = \frac{f_s}{f_0} = N'_0$$

L8.5

The most important message of this slide is how to work out N_0 and T_0 for a given sampling frequency fs (sampling period Ts = 1/fs).

The frequency step (for each successive frequency bin in the FFT result) = $1/T_0$. Given that the sampling frequency is fs, the number of samples you need is:

Number of samples: $N_0 = T_0 * fs$

Frequency step: $f_0 = 1/ T_0 = fs/ N_0$

For example, if fs = 10000Hz and we have a window width T_0 of 0.1sec, N_0 is 1000.

The frequency step is therefore 10Hz because in taking the DFT we pretend that the sampled signal in the T_0 window is repeated forever. Since the period of this periodic signal is T_0. When we take the DFT of this periodic signal, we get the Fourier coefficient for the fundamental frequency and all the harmonics.

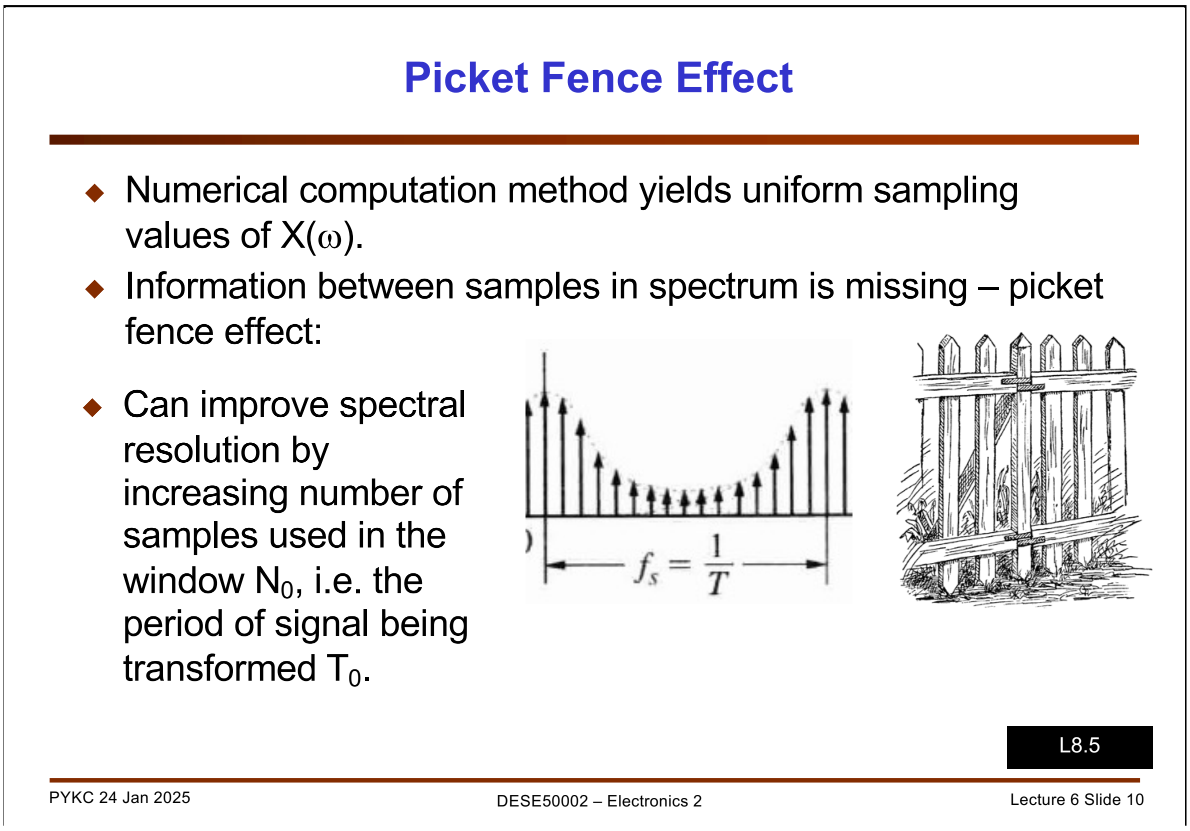
FFT is just a fast algorithm to computer the DFT coefficients. How the FFT algorithm works is outside the scope of this module. However, it is important to know this.

The complexity of DFT algorithm for a N point transform, which is also called the “Order” of the algorithm, is proportional to N^2 (or it is written as $O(N^2)$.

The complexity for an N point FFT is $O(Nlog_2N)$.

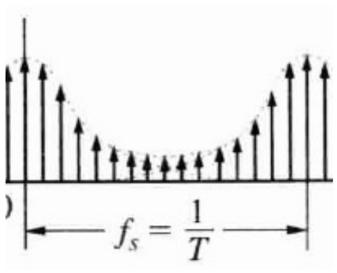
If N=1024, FFT is 100 times faster than DFT. DFT complexity = 1024 x 1024 $\approx 10^6$.

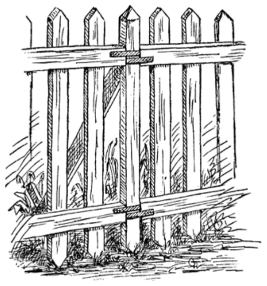
FFT complexity = 1024 x 1024 $\approx 10^4$.



Picket Fence Effect

- Numerical computation method yields uniform sampling values of $X(\omega)$.
- Information between samples in spectrum is missing – picket fence effect:
- Can improve spectral resolution by increasing number of samples used in the window N_0, i.e. the period of signal being transformed T_0.





L8.5

In other words, when we perform DFT, we create a periodic signal using a windowed version of the original (long) signal. Because of this action of constructing of a periodic signal from the original portion of signal, we get a spectrum with discrete frequency bins. This is in contrast with Fourier transform, which produces a continuous frequency spectrum (with ALL frequencies).

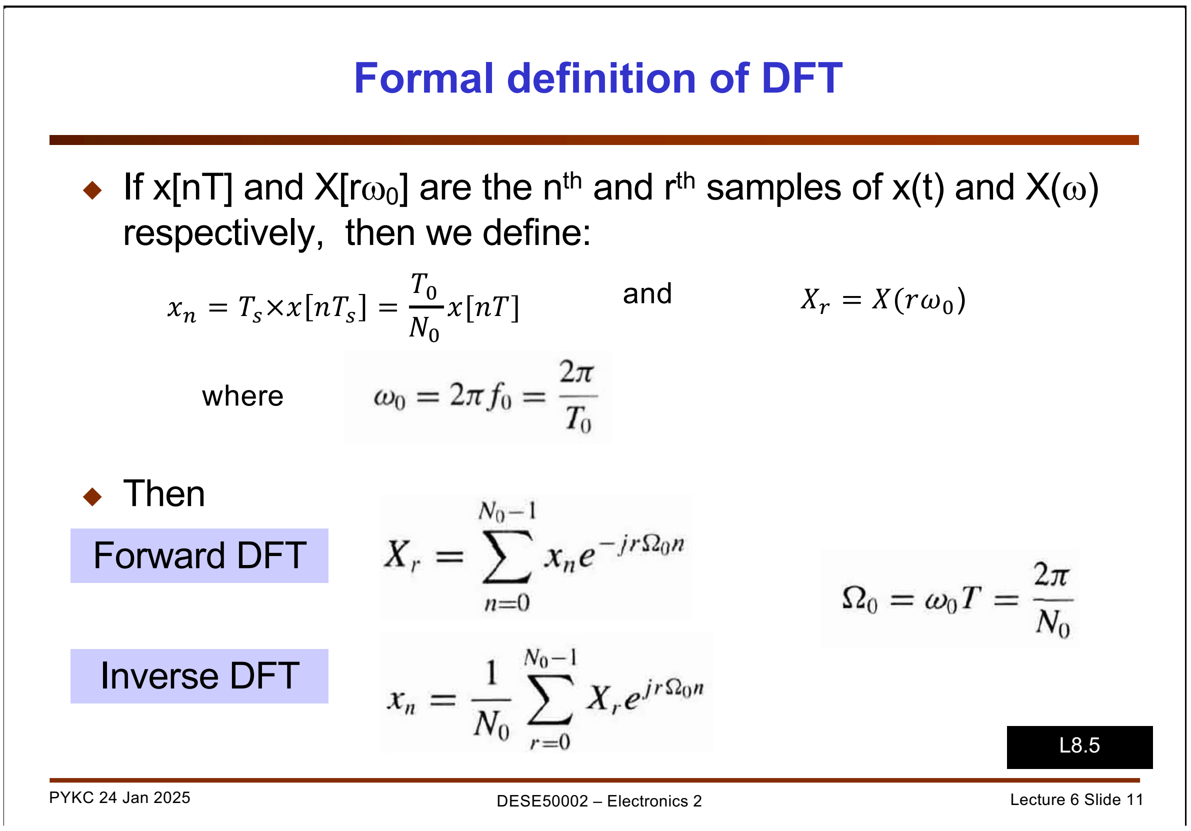
DFT is a computer based algorithm, therefore everything must be discrete.

The previous slide shows that the effect of DFT is to SAMPLE the continuous spectrum from the Fourier transform, and the rate of sampling (in frequency) is given by the frequency step $1/T_0 = fs/N_0$.

It is helpful to consider DFT as looking at the spectrum of the original signal through a picket fence. You only see samples of the spectrum at discrete frequencies. It also means that you may miss peak components if they fall between the sampling points (between the fence gaps and blocked by the fence posts).

Let us consider an example where we sample a signal at 8000 Hz, and we extract 1000 samples to analyse using DFT (actually we used FFT, the fast version of DFT in Matlab).

The frequency resolution (step) is 8000Hz/ 1000 = 8Hz. The DFT algorithm will generate a spectrum between -4000Hz and +4000Hz in steps of 8Hz. This yields 1000 frequency bins. In Matlab code, magnitude[1] = 0Hz or dc value. magnitude[2] = 8Hz bin etc.. Now if you use the tuning fork app to generate a 1000Hz tone, you will see a clear spike in magnitude[126] (exactly 125 x8 = 1000Hz). However, if you use a 1100Hz tone, the signal falls between magnitude[138] and magnitude[139] (bin 138.5 = 1100/8 + 1). So what you should see is the energy shared in the two neighbouring frequency bins.



Formal definition of DFT

- If x[nT] and X[rω$_0$] are the nth and rth samples of x(t) and X(ω) respectively, then we define:

$$x_n = T_s \times x[nT_s] = \frac{T_0}{N_0} x[nT] \qquad \text{and} \qquad X_r = X(r\omega_0)$$

where $\omega_0 = 2\pi f_0 = \frac{2\pi}{T_0}$

- Then

Forward DFT $$X_r = \sum_{n=0}^{N_0-1} x_n e^{-jr\Omega_0 n}$$

$$\Omega_0 = \omega_0 T = \frac{2\pi}{N_0}$$

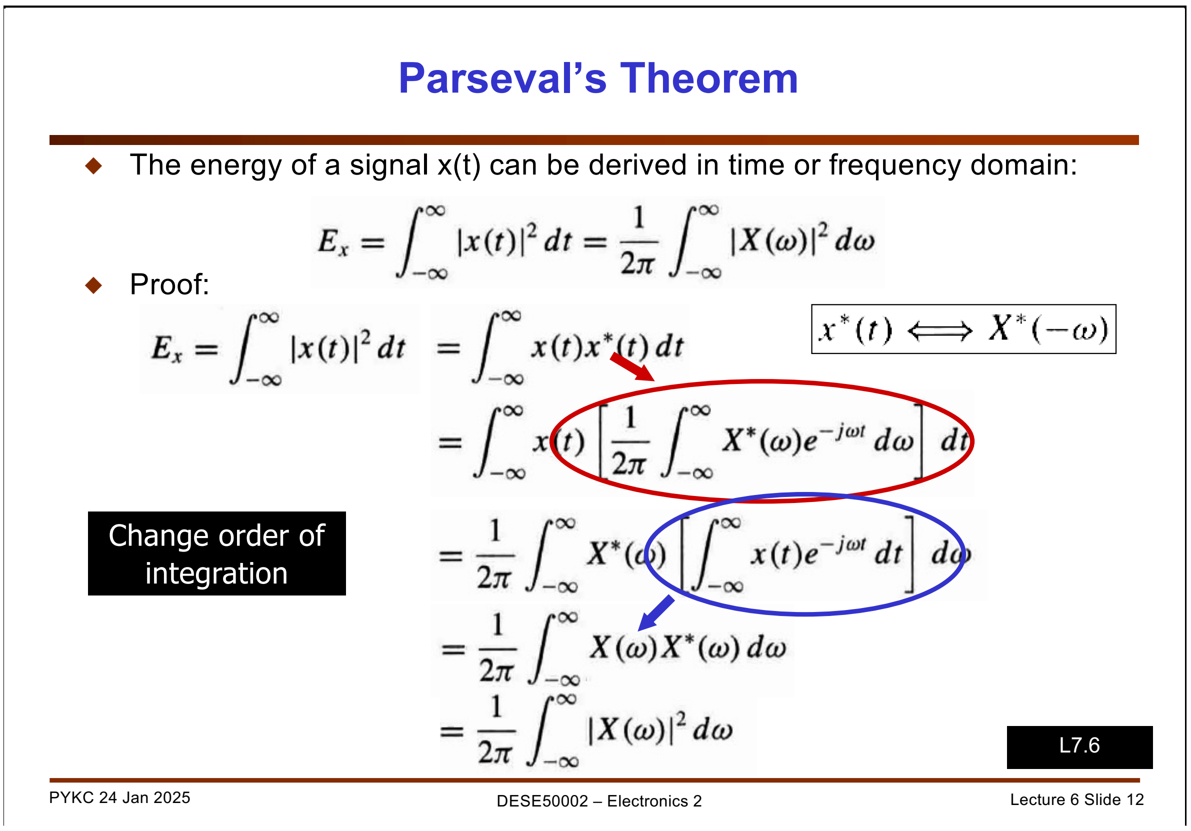
Inverse DFT $$x_n = \frac{1}{N_0} \sum_{r=0}^{N_0-1} X_r e^{jr\Omega_0 n}$$

L8.5

PYKC 24 Jan 2025 DESE50002 – Electronics 2 Lecture 6 Slide 11

Here is a formal mathematical definition of Discrete Fourier Transform. You are not really required to remember this. If you compare this with the formal definition of the Fourier transform (not the discrete version), they look very similar except that

1. x(t) now becomes x[nT] because samples are discrete. T_s is the sampling interval and $T_s = 1/f_s$.
2. ω_0 is the fundamental frequency as we repeat the extract signals every T_0. $\omega_0 = 2\pi f_0 = 2\pi/T_0$.
3. DFT produces results Xr, the amount of signal at the rth frequency bin, whose frequency is $r\Omega_0$, where Ω_0 is the discrete frequency step.



Parseval's Theorem

- The energy of a signal x(t) can be derived in time or frequency domain:

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

- Proof:

$$\boxed{x^*(t) \Longleftrightarrow X^*(-\omega)}$$

$$\begin{aligned}
E_x = \int_{-\infty}^{\infty} |x(t)|^2\, dt &= \int_{-\infty}^{\infty} x(t)x^*(t)\, dt \\
&= \int_{-\infty}^{\infty} x(t)\left[\frac{1}{2\pi}\int_{-\infty}^{\infty} X^*(\omega)e^{-j\omega t}\, d\omega\right] dt \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} X^*(\omega)\left[\int_{-\infty}^{\infty} x(t)e^{-j\omega t}\, dt\right] d\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)X^*(\omega)\, d\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} |X(\omega)|^2\, d\omega
\end{aligned}$$

Change order of integration

L7.6

PYKC 24 Jan 2025 DESE50002 – Electronics 2 Lecture 6 Slide 12

The next three slides are about energy of a signal.

We have seen before in Lecture 1 and Lab 2 Exercise 4 that we can compute the energy of a signal by integrating the square of the signal x^2(t) over the duration where the signal is non-zero.

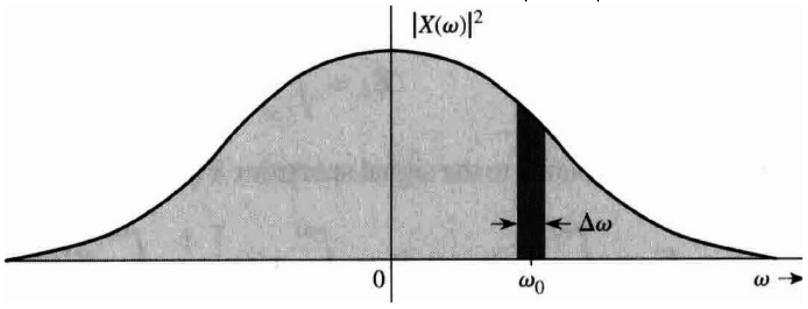
Parseval's Theorem basically says that you can do this energy calculation equally well in the frequency domain. The important result here is the equation:

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

We can compute energy of the signal by integrating over all the spectral frequencies ($\omega = -\infty$ to $\omega = +\infty$) the square of the magnitude of the spectrum. The 1/2π is a scaling factor to account for the fact that we using radians/sec as frequency and not Hz (cycles/sec).

Energy Spectral Density of a signal

- Total energy is area under the curve of $|X(\omega)|^2$ vs ω (divided by 2π).



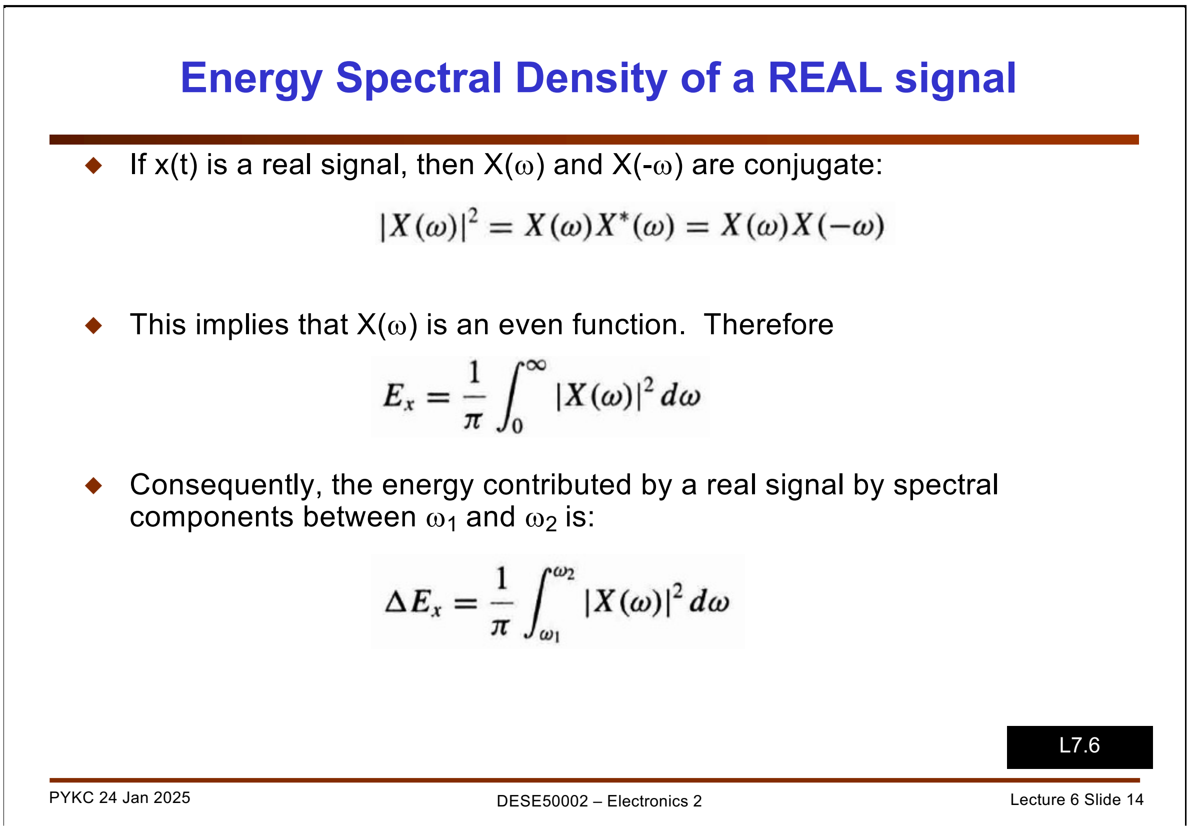
- The energy over a small frequency band Δω (Δω→0) is:

$$\Delta E_x = \frac{1}{2\pi}|X(\omega)|^2\,\Delta\omega = \underbrace{|X(\omega)|^2}_{\text{Energy spectral density (per unit bandwidth in Hz)}}\,\Delta f \qquad \frac{\Delta\omega}{2\pi} = \Delta f \text{ Hz}$$

L7.6

Therefore we can view the plot of |X(ω)|^2 as the energy spectral density of the signal. We can then ask the question: “how much energy is between two frequencies?” by integrating under the curve as shown.

We can this energy spectral density because the unit can be seen as energy per unit bandwith (in Hz).



Energy Spectral Density of a REAL signal

- If x(t) is a real signal, then X(ω) and X(-ω) are conjugate:

$$|X(\omega)|^2 = X(\omega)X^*(\omega) = X(\omega)X(-\omega)$$

- This implies that X(ω) is an even function. Therefore

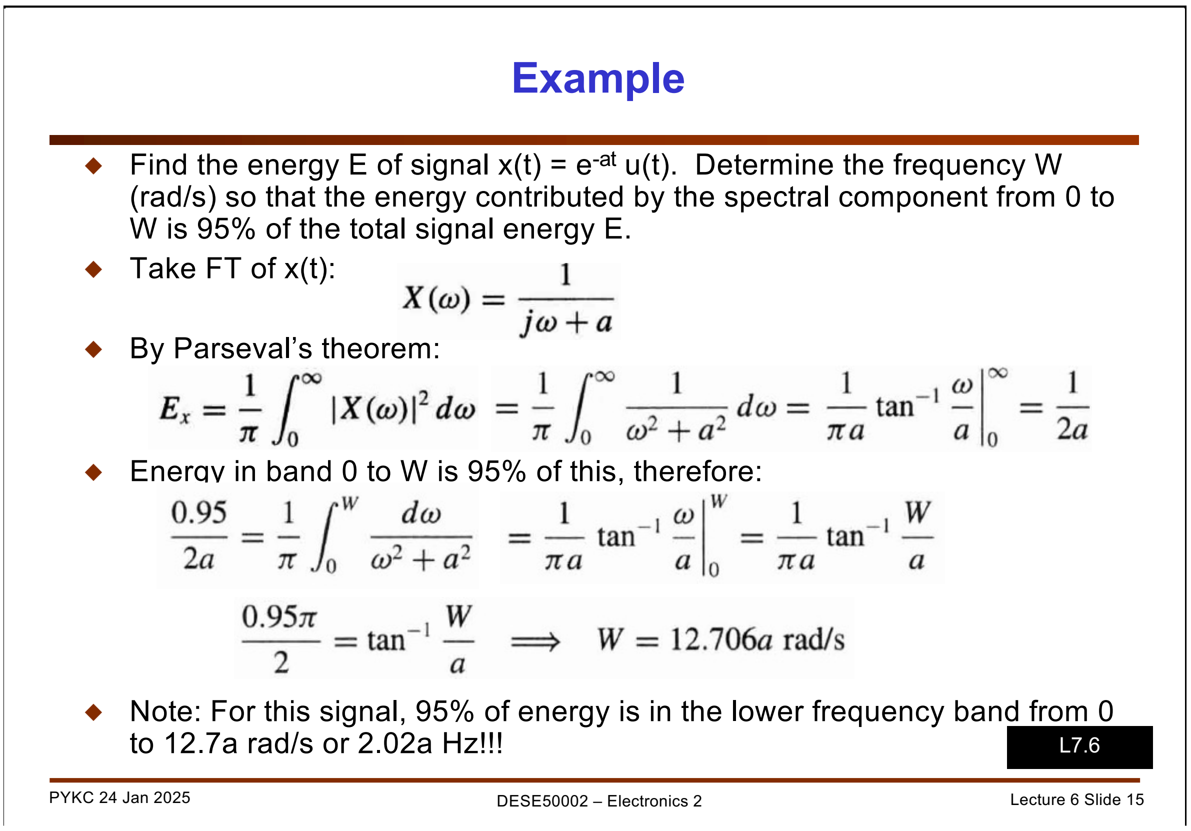
$$E_x = \frac{1}{\pi}\int_0^\infty |X(\omega)|^2\,d\omega$$

- Consequently, the energy contributed by a real signal by spectral components between ω_1 and ω_2 is:

$$\Delta E_x = \frac{1}{\pi}\int_{\omega_1}^{\omega_2} |X(\omega)|^2\,d\omega$$

L7.6

For real signals, it can be shown that the spectrum is symmetrical about 0, i.e. X(ω) = X(-ω). Therefore we can simplify the calculation of energy between two frequencies ω_1 and ω_2 as shown.



Example

- Find the energy E of signal $x(t) = e^{-at}u(t)$. Determine the frequency W (rad/s) so that the energy contributed by the spectral component from 0 to W is 95% of the total signal energy E.
- Take FT of x(t):

$$X(\omega) = \frac{1}{j\omega + a}$$

- By Parseval's theorem:

$$E_x = \frac{1}{\pi}\int_0^{\infty} |X(\omega)|^2\, d\omega = \frac{1}{\pi}\int_0^{\infty} \frac{1}{\omega^2 + a^2}\, d\omega = \frac{1}{\pi a}\tan^{-1}\frac{\omega}{a}\bigg|_0^{\infty} = \frac{1}{2a}$$

- Energy in band 0 to W is 95% of this, therefore:

$$\frac{0.95}{2a} = \frac{1}{\pi}\int_0^{W} \frac{d\omega}{\omega^2 + a^2} = \frac{1}{\pi a}\tan^{-1}\frac{\omega}{a}\bigg|_0^{W} = \frac{1}{\pi a}\tan^{-1}\frac{W}{a}$$

$$\frac{0.95\pi}{2} = \tan^{-1}\frac{W}{a} \implies W = 12.706a \text{ rad/s}$$

- Note: For this signal, 95% of energy is in the lower frequency band from 0 to 12.7a rad/s or 2.02a Hz!!!

L7.6

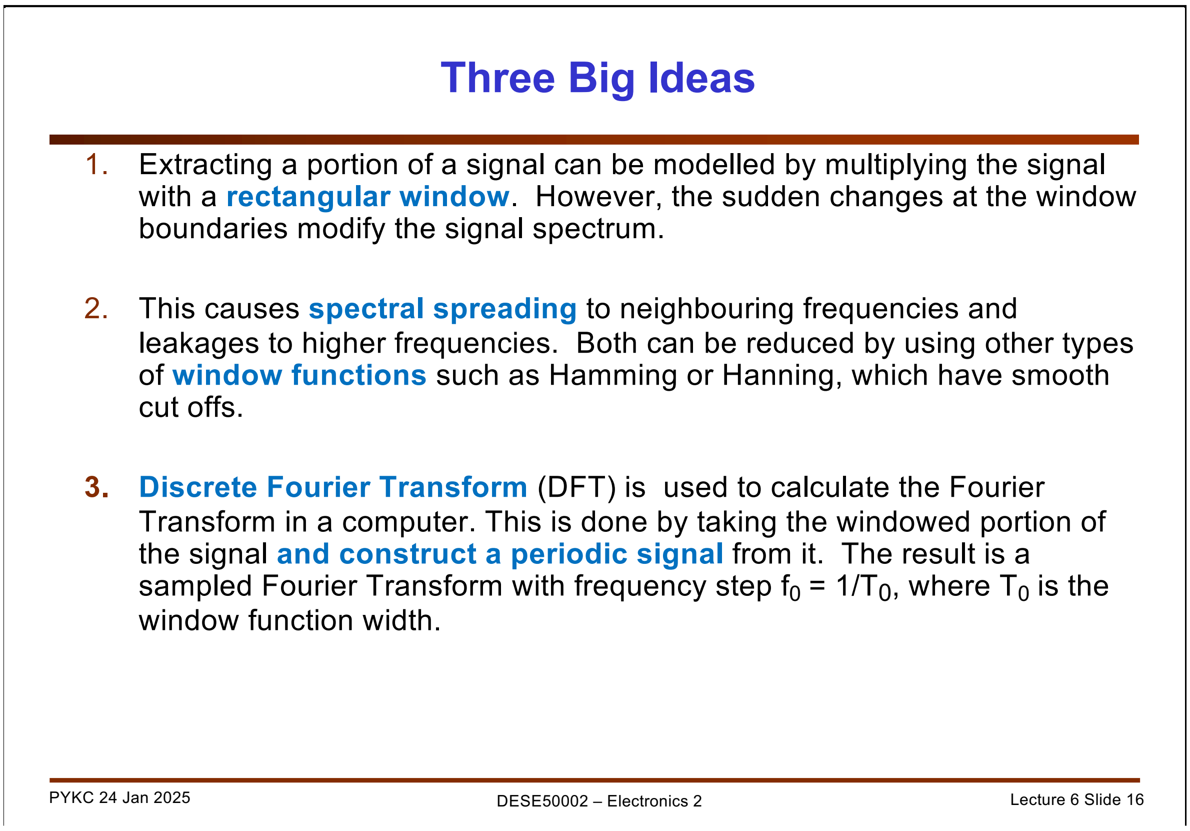
Here is an example of how to use Parseval's Theorem to provide useful results.

We are interested in the bandwidth of a signal which contain 95% of the energy. The signal in question is $x(t) = e^{-at}u(t)$.

This is done by performing FT on x(t) using the FT table.

Using Parseval's Theorem we can compute the integral and calculate the total energy of the signal Ex.

The rest is pretty straightforward.



Three Big Ideas

1. Extracting a portion of a signal can be modelled by multiplying the signal with a **rectangular window**. However, the sudden changes at the window boundaries modify the signal spectrum.

2. This causes **spectral spreading** to neighbouring frequencies and leakages to higher frequencies. Both can be reduced by using other types of **window functions** such as Hamming or Hanning, which have smooth cut offs.

3. **Discrete Fourier Transform** (DFT) is used to calculate the Fourier Transform in a computer. This is done by taking the windowed portion of the signal **and construct a periodic signal** from it. The result is a sampled Fourier Transform with frequency step $f_0 = 1/T_0$, where T_0 is the window function width.

Here are three things that worth remembering and understand.