The Discrete Fourier Transform

Textbook pages 461-471; 476-482; 500-507

Dr D. Laurenson

3rd September 2020

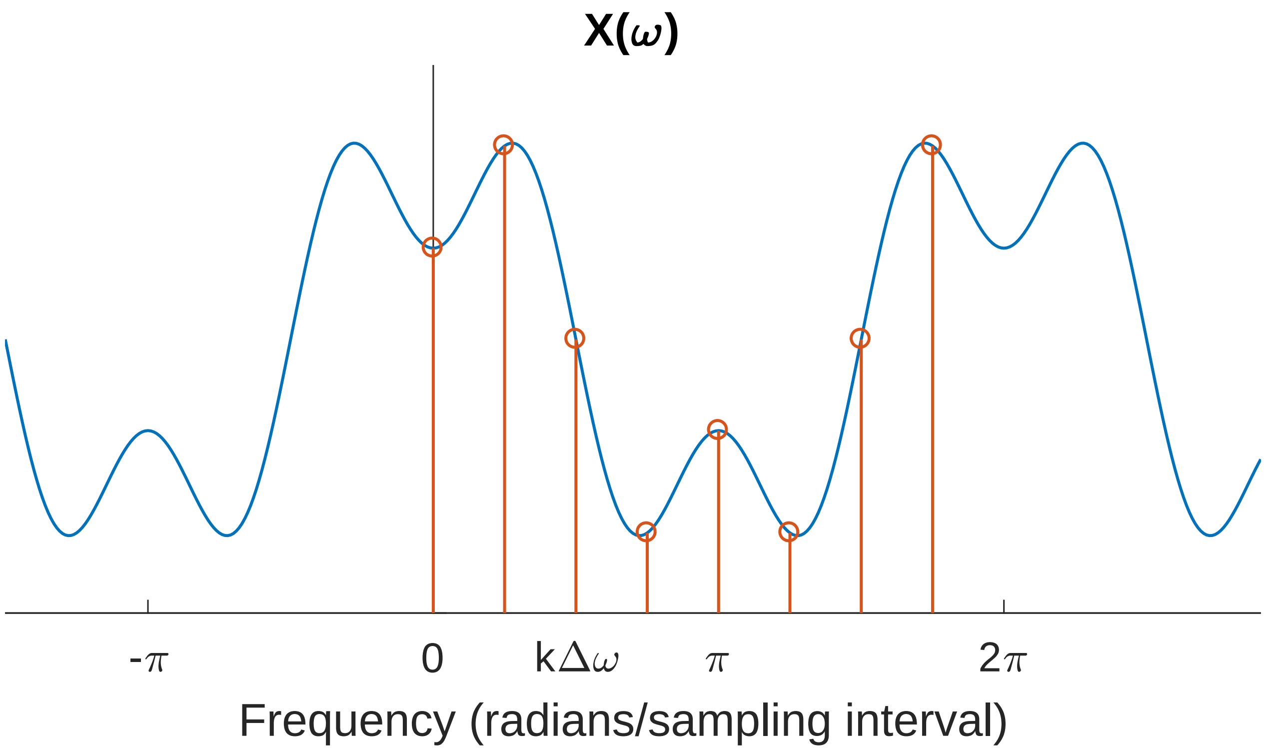
Frequency-Domain Sampling

The derivation presented in this section is an alternative to that presented in the course textbook. Both are equally valid, and both arrive at the same conclusion for the same reason. It is not a requirement that you understand both, however you may find one of the two explanations easier to understand than the other.

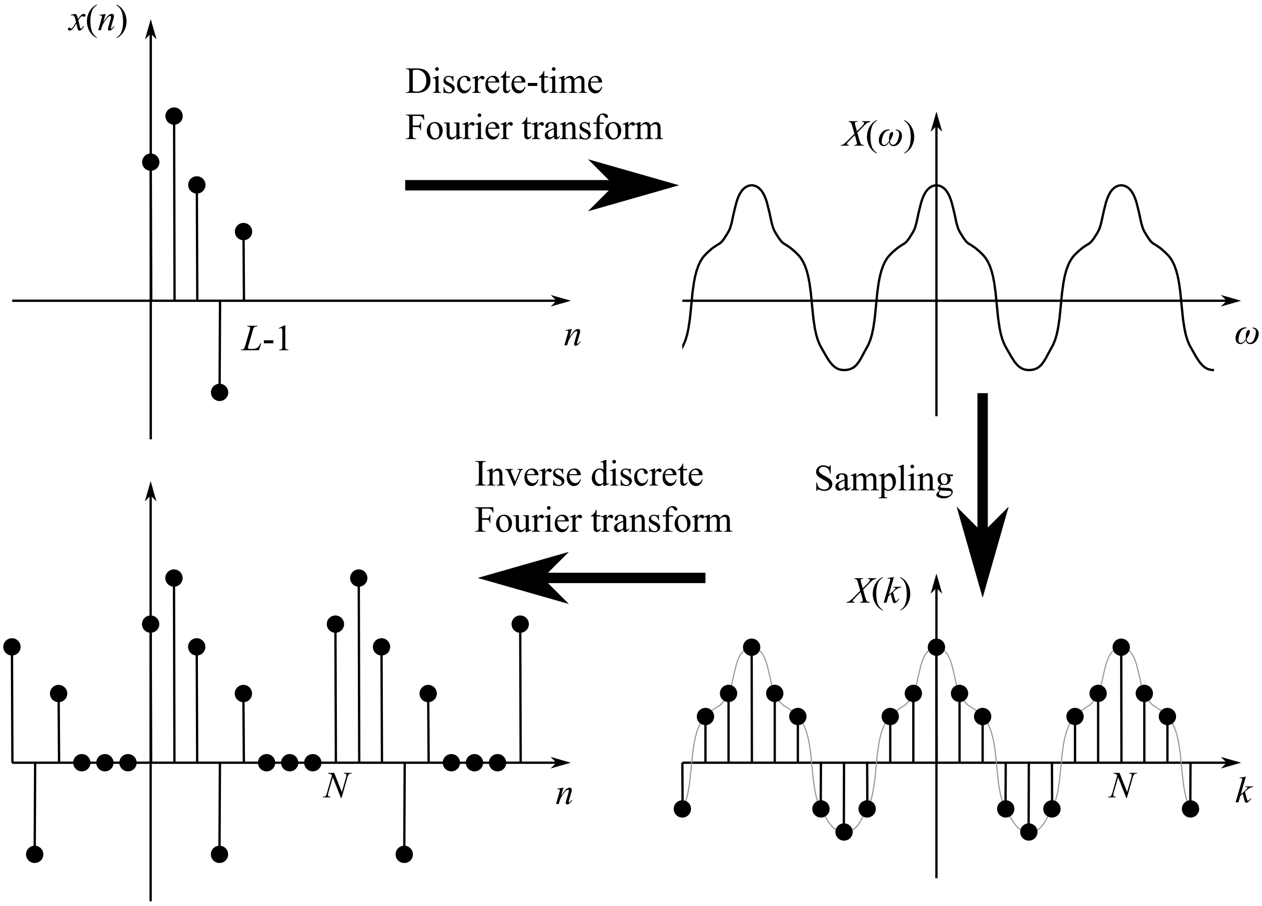
Starting with a finite set of samples of a data sequence, $x(n)$, with $0 \le n < L$, the frequency domain can be written as $X(\omega) = \sum_{n=0}^{L-1} x(n)e^{-j\omega n}$. The transform is continuous in the frequency domain as the signal is aperiodic in the time domain. However, the transform is periodic in the frequency domain as the signal is sampled in the time domain. It is desirable to represent the frequency domain by a finite number of samples of $X(\omega)$, taken over one period of the frequency domain, as this can be readily represented in a digital processor.

Sampling over one full period of the frequency domain with N samples results in a sample spacing, $\Delta\omega$, of $\Delta\omega = \frac{2\pi}{N}$. Thus samples are evaluated as:

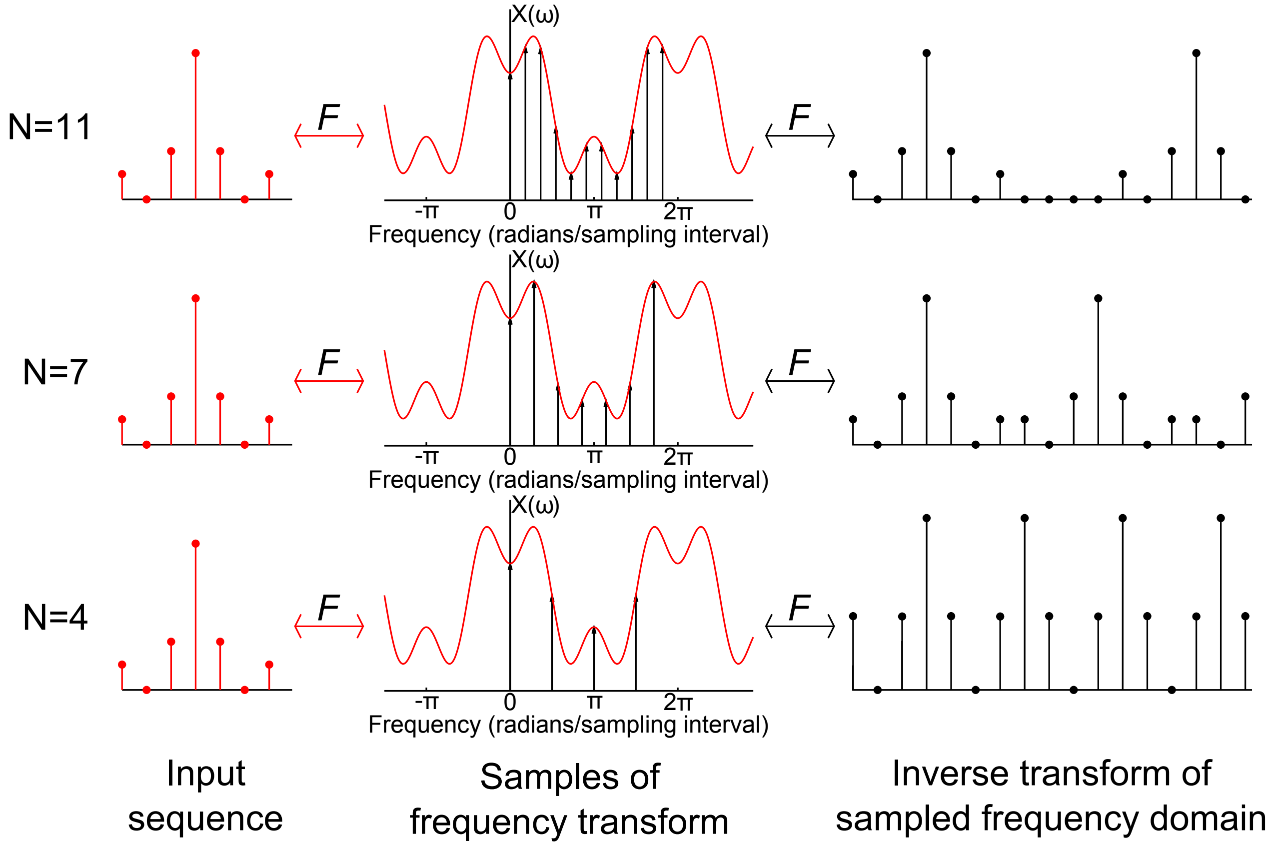
$$X\left(\frac{2\pi k}{N}\right) = \sum_{n=0}^{L-1} x(n)e^{-j2\pi kn/N}$$



In order to determine an appropriate value of N, we need to consider the effect of sampling. Consider, first, the effect of sampling in the time domain. A discrete-time signal has a periodic transform, where the period is inversely related to the sample spacing in time, Δt. Formally, $f_s = \frac{1}{\Delta t}$, or equivalently, $\omega_s = \frac{2\pi}{\Delta t}$. Because of the duality relationship, if the frequency domain is sampled, then the time domain becomes periodic. The period of the time domain is inversely related to the spacing of samples in frequency, $\Delta\omega = 2\pi\Delta f$, thus the period, T, is given by $T = N\Delta t$. In other words, if an inverse transform is taken, the resulting period in the time domain is N samples.



Starting with the case that $N > L$, when an inverse transform of $X\left(\frac{2\pi k}{N}\right)$ is taken, the periodic repetitions of the original samples will be separated by a number of zero samples. As N is decreased, the spacing of the periodic repetitions reduces, to the point when $N = L$, and there are no zero samples between the periodic repetitions. Reducing N further results in an overlap between periodic repetitions, and time domain aliasing occurs. The figure below shows an example where the input sequence is seven samples long, $L = 7$. The continuous frequency transform is shown, with N equally spaced samples over one period. The inverse transform of the samples is shown in the final column. Where $N < L$, time domain aliasing can clearly be observed.



In this figure, the input sequence (shown in red) is sampled and finite in length. It has a continuous frequency transform (also shown in red). By sampling this transform with n samples, as shown in black, results in a periodic time domain signal, shown on the right in black. Where the samples are identical to the input sequence, no information has been lost by the sampling process ($N = 11$ and $N = 7$). However, when too few samples are taken, the values of the original input sequence are lost ($N = 4$), meaning that the sampled frequency domain cannot properly represent the original input sequence.

The Discrete Fourier Transform

Textbook pages 461-471; 476-482; 500-507

Dr D. Laurenson

3rd September 2020

Discrete Fourier Transform (DFT)

In order to minimise the computational complexity of any analysis in the frequency domain, it can be useful to minimise the number of samples of the discrete-time Fourier transform that need to be taken. (Note, this effectively means that we only evaluate the transform at discrete frequencies, and never compute the complete continuous frequency domain transform). It is common to sample the frequency domain at equally spaced points, which gives rise to the discrete Fourier transform (DFT).

Definition of DFT

The minimum number of samples, to avoid loss of information, is $N = L$.

It is possible to choose any value of $N \geq L$, and there are applications where this is desirable. We will cover more on this when we explore the effect of zero padding.

The discrete Fourier transform (DFT) is then defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \tag{1.18}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \tag{1.19}$$

The transform is periodic in time and in frequency with a period of N samples.

As an example, consider a sequence:

$$x(n) = \begin{cases} 1 & ; 0 \leq n < L \\ 0 & ; n \geq L \end{cases}$$

Taking an N point DFT of this sequence, where $N \geq L$, gives:

$$X(k) \quad = \quad \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} = \sum_{n=0}^{L-1} e^{-j2\pi nk/N}$$

Using the formula for a geometric series, we obtain:

$$X(k) = \frac{1 - e^{-j2\pi kL/N}}{1 - e^{-j2\pi k/N}}$$

To simplify this expression, the simplest, but least obvious, way is to identify that the numerator and denominator can be re-expressed in terms of the Euler expansion of sin. This expands the fraction as follows:

$$\begin{aligned} X(k) &= \frac{e^{-j\pi kL/N}}{e^{-j\pi k/N}} \frac{e^{j\pi kL/N} - e^{-j\pi kL/N}}{e^{j\pi k/N} - e^{-j\pi k/N}} \\ &= e^{-j\pi k(L-1)/N} \frac{2j \sin(\pi kL/N)}{2j \sin(\pi k/N)} \\ &= e^{-j\pi k(L-1)/N} \frac{\sin(\pi kL/N)}{\sin(\pi k/N)} \end{aligned}$$

The resulting magnitude and phase plots are shown below:[1]

[1] Selected figures taken from "Digital Signal Processing, New International Edition/4th", Proakis & Manolakis, ©Pearson Education Limited, 2014. ISBN: 978-1-29202-573-5

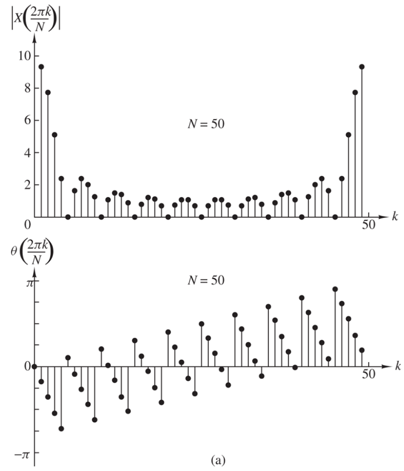


Figure 7.1.6 Magnitude and phase of an N-point DFT in Example 7.1.2; (a) $L = 10$, $N = 50$; (b) $L = 10$, $N = 100$.

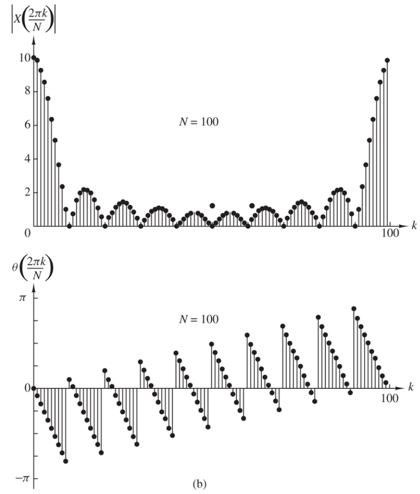


Figure 7.1.6 **Figure** *continued*

The Discrete Fourier Transform

Textbook pages 461-471; 476-482; 500-507

Dr D. Laurenson

3rd September 2020

Properties of the DFT

Linearity	
If $x_1(n) \overset{\text{DFT}}{\longleftrightarrow} X_1(k)$ and $x_2(n) \overset{\text{DFT}}{\longleftrightarrow} X_2(k)$	$a_1x_1(n) + a_2x_2(n) \overset{\text{DFT}}{\longleftrightarrow} a_1X_1(k) + a_2X_2(k)$
Symmetry	
If $x_p(n) = x_p(-n)$	$X(k) = X(-k)$
If $x_p(n) = -x_p(-n)$	$X(k) = -X(-k)$
Periodicity	
For an N point DFT	$x_p(n+N) = x_p(n)$
	$X(k+N) = X(k)$

Because of periodicity, the symmetry properties can also be expressed in terms of the input signal as follows

Symmetry	
If $x(n)=x(N-n)$	$X(k)=X(-k)=X(N-k)$
If $x(n)=-x(N-n)$	$X(k)=-X(-k)=-X(N-k)$

For real valued inputs, $\Re\mathfrak{e}(x(n)) = x(n)$, the transform is complex conjugate symmetric: $X(k) = X^*(-k) = X^*(N-k)$, and $|X(k)| = |X(-k)| = |X(N-k)|$.

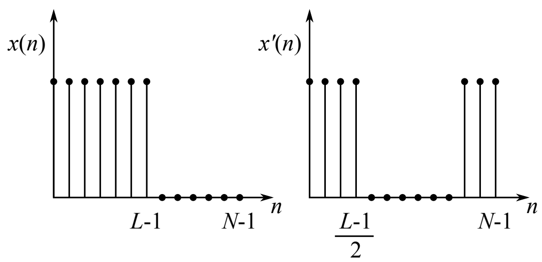
A *cyclic* shift in time is equivalent to multiplication by a complex phase in frequency. I.e. if $x'(n) = x(n{-}l, \text{modulo } N) = x((n-l))_N$, then $X'(k) = e^{-j2\pi kl/N}X(k)$.

The fast Fourier transform (FFT) is an efficient method of computing a DFT if N is a power of 2.

As an example, consider the sequence:

$$x(n) = \begin{cases} 1 & ; 0 \le n < L \\ 0 & ; n \ge L \end{cases}$$

where L is an odd number. We will examine a cyclic shift of $(L-1)/2$ samples.



We know that the DFT is given by:

$$X(k) = e^{-j\pi k(L-1)/N}\frac{\sin(\pi kL/N)}{\sin(\pi k/N)}$$

If the signal is cyclically shifted to the *left* by $(L-1)/2$ samples, then $l = -(L-1)/2$, and

$$X'(k) = e^{j2\pi k(L-1)/2N}X(k) = \frac{\sin(\pi kL/N)}{\sin(\pi k/N)}$$

This is the transform of a cyclically symmetric real even signal, where

$$\begin{aligned} x'(n) &= x\left(\left(n + \frac{L-1}{2}\right)\right)_N \\ &= \begin{cases} x\left(n + \frac{L-1}{2}\right) & ; n < N - (L-1)/2 \\ x\left(n - N + \frac{L-1}{2}\right) & ; n \ge N - (L-1)/2 \end{cases} \end{aligned}$$

Choosing a value for N

Given the above, because the frequency domain imposes a periodicity in the time domain when finding the inverse discrete Fourier transform (IDFT), $N \geq L$ to avoid corruption of the signal. To avoid unnecessary computation, very often the minimum value of N should be used, i.e. $N = L$. However, due to an efficient method of computation, called the fast Fourier transform, in practice N is rounded up to the nearest power of 2. So, if $L = 1000$, then the most efficient choice of N is 1024. Samples from L to $N-1$ in the input are assumed to be zero, and the process called zero padding.

The Python function to compute a fast Fourier transform in the numpy package is `np.fft()`, which will accept any positive value for N, however it computes the transform more quickly if N is a power of 2 compared to other values of N. If the specified FFT length, N, is larger than the signal passed in to `np.fft()`, then Python will zero-pad up to length N before performing the transform.

The Discrete Fourier Transform

Textbook pages 461-471; 476-482; 500-507

Dr D. Laurenson

8th October 2020

1 Frequency analysis of signals using the DFT

Returning to the introduction to this module, two steps were identified: windowing of the input signal; and sampling in the frequency domain. We now turn our attention to the first of these to assess, and control, the impact of this process.

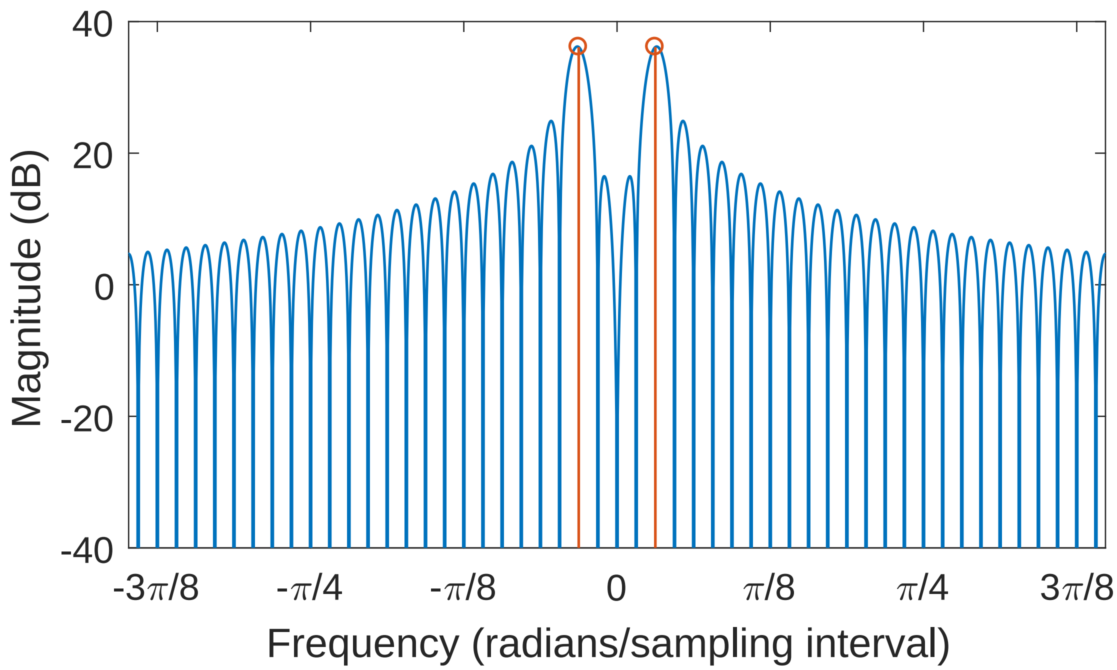
Consider a previous example, where the input signal is

$$x(n) = \cos(n\pi/32)$$

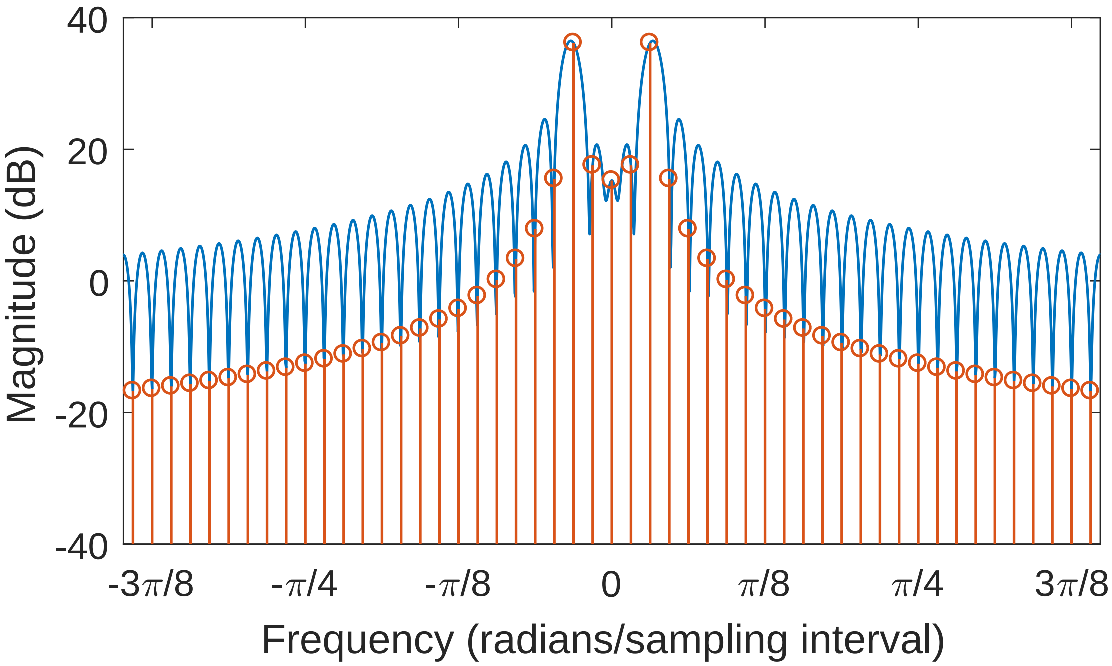
and the window

$$w(n) = \begin{cases} 1 & ;0 \leq n < 128 \\ 0 & ;\text{otherwise} \end{cases}$$

It was established that if $a(n) = x(n)w(n)$, then $A(\omega) = X(\omega) * W(\omega)$. Applying this to the DFT it is readily seen that this process is equivalent to selecting 128 samples of the input signal. Thus, the DFT will be a sampled version of $A(\omega)$ and *not* $X(\omega)$ as initially desired. In this case, if $N = 128$, then the samples of $A(\omega)$ appear as below:



Note how the sample points are spaced by $\Delta\omega = 2\pi/N$, and also that this spacing is the same as the spacing of nulls of the window function $W(\omega)$. This is because the length of the window in the time domain, which corresponds to how "narrow" it is in the frequency domain, is the same as the number of samples in the transform. Thus, in this case, $A(\omega)$ has only two non-zero samples, corresponding to the positive and negative phasors that define the input cosine waveform. It would appear, therefore, that the windowing step has had no effect. However, if the input signal is changed to have a slightly higher frequency ($x(n) = \cos(1.05n\pi/32)$), then a very different result is obtained.



Here, the sample points are taken in the same places as before, and the spacing of the nulls in the window are the same as before, however it is obvious that there are many more non-zero samples due to the different centre frequency of the input signal. The production of many more non-zero samples for a single frequency component is termed *leakage*, as the energy of the signal "leaks" into adjacent bins.

Ideally we would like an analysis for any single frequency input to give an output that represents this with a single sample (or pair of samples for a real valued input). Thus, additional steps are required in order to change the result into one that is closer to what we desire. In order to do this, then we need to re-examine the steps that were taken to produce the DFT.

Sampling in frequency We cannot change the need to take samples, however we could change the spacing of samples. One simple way to do this is to increase the value of N using zero padding, which is equivalent to adding $N-L$ zeros to the input and then performing an N-point DFT. Taking this step will result in more samples, and the samples being spaced closer together. However, it will still be $A(\omega)$ that is being sampled, thus this will result in more non-zero samples than in the first case, so this doesn't solve the problem.

Another way of changing the spacing of samples is to acquire more input samples, i.e. make L, as well as N, larger. In this particular example, if we were to make $N = L = 1280$ it turns out that we would once again have only two non-zero samples. The reason is that this is the first value of N and L for which the spacing of nulls from $W(\omega)$, combined with the cosine frequency, result in alignment with the position of samples. Obviously this is *not* useful as we, by definition, do not know the frequency components of a signal that we wish to analyse before we analyse it! (Also it is very bad practice to have an analysis method whose parameters depend on the input signal).

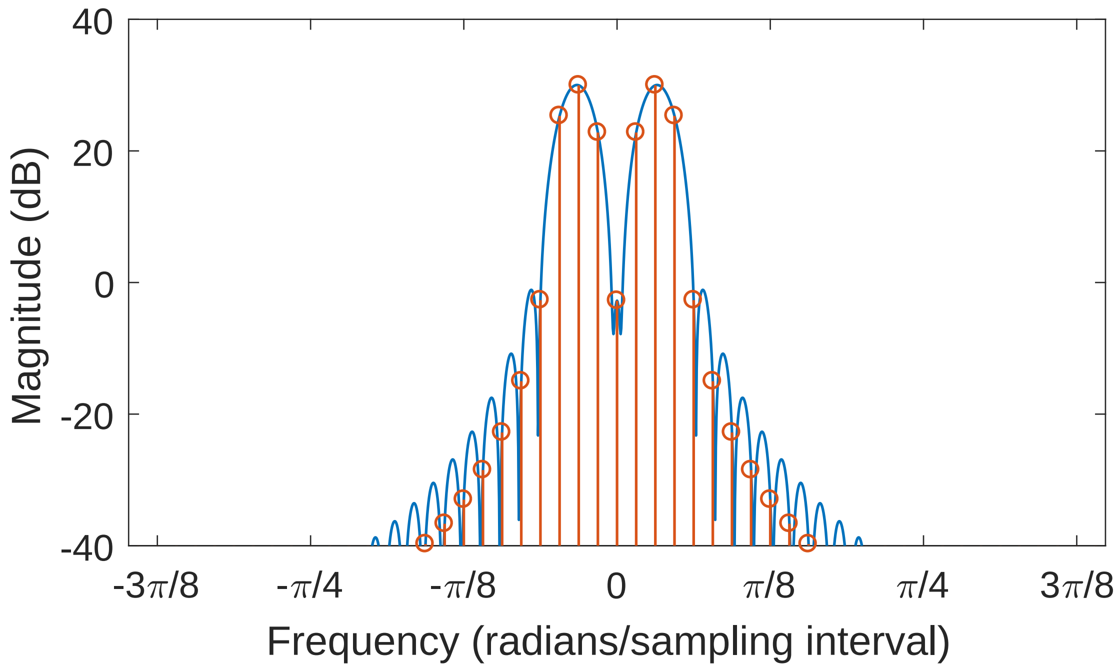
Windowing As an alternative, the process of windowing can be explored. If the shape of $W(\omega)$ is changed, then the nature of the leakage can be changed. Note that it is not possible to remove leakage using this technique, but it is possible to control how it affects the transform. Changing $W(\omega)$ is easily done by changing $w(n)$. So, instead of simply taking L inputs, and putting them into a DFT, the input values are first scaled by $w(n)$, and then the DFT is applied. This is a particularly good approach as it is independent of the frequencies contained in the input signal, so can be applied uniformly across all analysis.

A family of raised cosine windows can be defined by $w(n) = (1-\alpha) - \alpha\cos\left(\frac{2\pi n}{L-1}\right)$ where $0 \le n < L$, L is the number of input samples being considered where $L \le N$, and N is the number of samples in the DFT. The Hann (or Hanning) window is obtained when $\alpha = 0.5$. The window function can be computed using the discrete-time transform to give the following result:

$$
\begin{aligned}
W(\omega) &= \sum_{n=0}^{L-1} w(n) e^{-j\omega n} \\
&= e^{-j\omega(L-1)/2}\left\{(1-\alpha)\frac{\sin(\omega L/2)}{\sin(\omega/2)}\right. \\
&\qquad \left. + \frac{\alpha}{2}\frac{\sin\left(\frac{(\omega(L-1)-2\pi)L}{2(L-1)}\right)}{\sin\left(\frac{\omega(L-1)-2\pi}{2(L-1)}\right)} + \frac{\alpha}{2}\frac{\sin\left(\frac{(\omega(L-1)+2\pi)L}{2(L-1)}\right)}{\sin\left(\frac{\omega(L-1)+2\pi}{2(L-1)}\right)}\right\}
\end{aligned}
$$

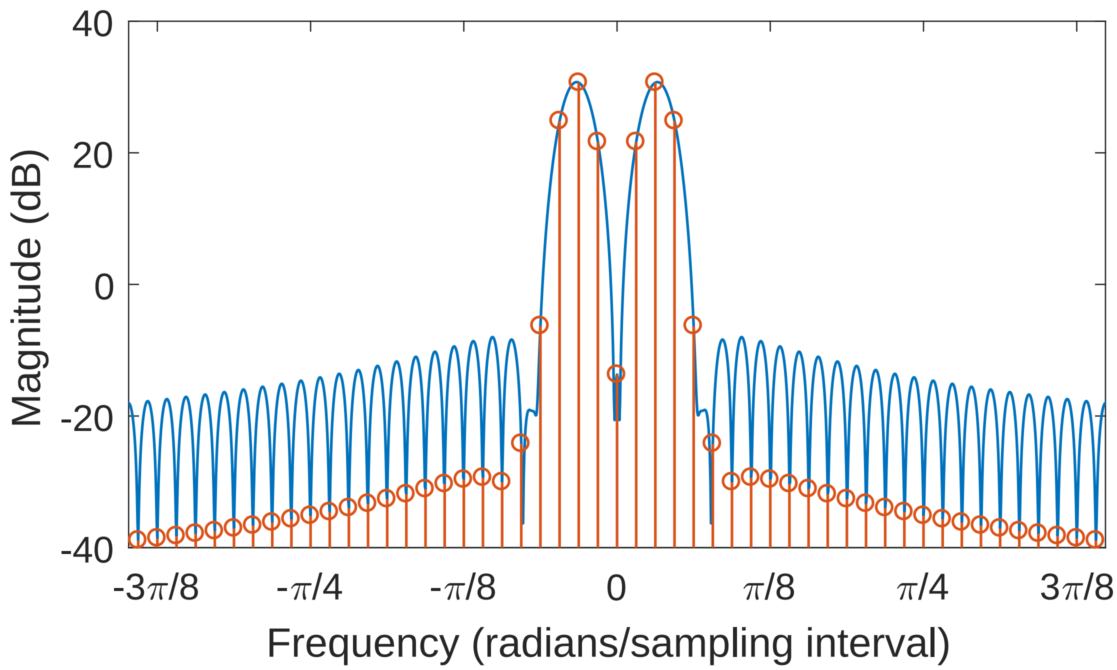
The window is applied by multiplying the L samples of $x(n)$ by the L samples of $w(n)$. This is then used as the input to the N-point DFT. Thus, the input sequence to the DFT is $\{w(0)x(0), w(1)x(1), w(2)x(2), \ldots, w(L-1)x(L-1)\}$, which will be zero-padded, by appending 0's to the end of the sequence, if $L < N$. Note that as the samples are being scaled, the signal power, when estimated from the frequency domain, will be scaled. The effect of the scaling from the window can be corrected by dividing the product of $x(n)$ and $w(n)$ by $\sqrt{\left(\sum_{n=0}^{L-1}(w(n))^2\right)/N}$.

Applying a Hann window results in the following transform:



The leakage has not been removed by application of the window, but because the shape of the window in the frequency domain is different to the shape of the rectangular window (i.e. the implicit window when only L samples of the potentially infinite length input are selected), the nature of the leakage changes. It is clear that around the frequency corresponding to the input frequency that the window has increased the magnitude of the transform, whilst at more distant frequencies, the magnitude has been significantly reduced.

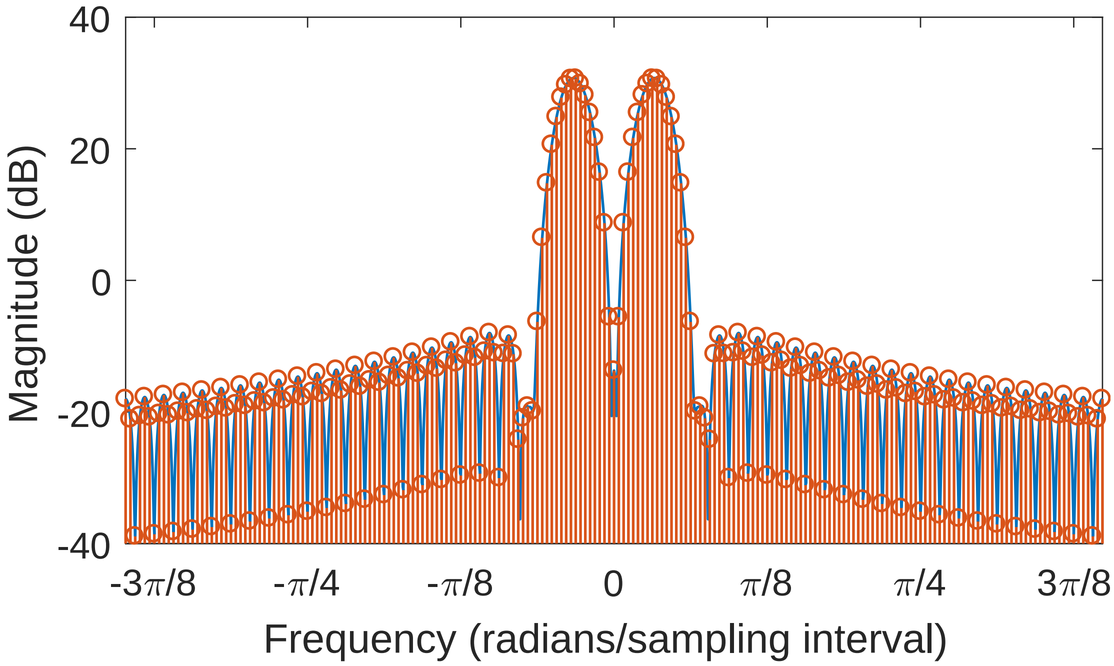
Another well known window, the Hamming window, can be obtained by setting $\alpha = 0.46$. Applying this window, instead of the Hann window, produces the following result:



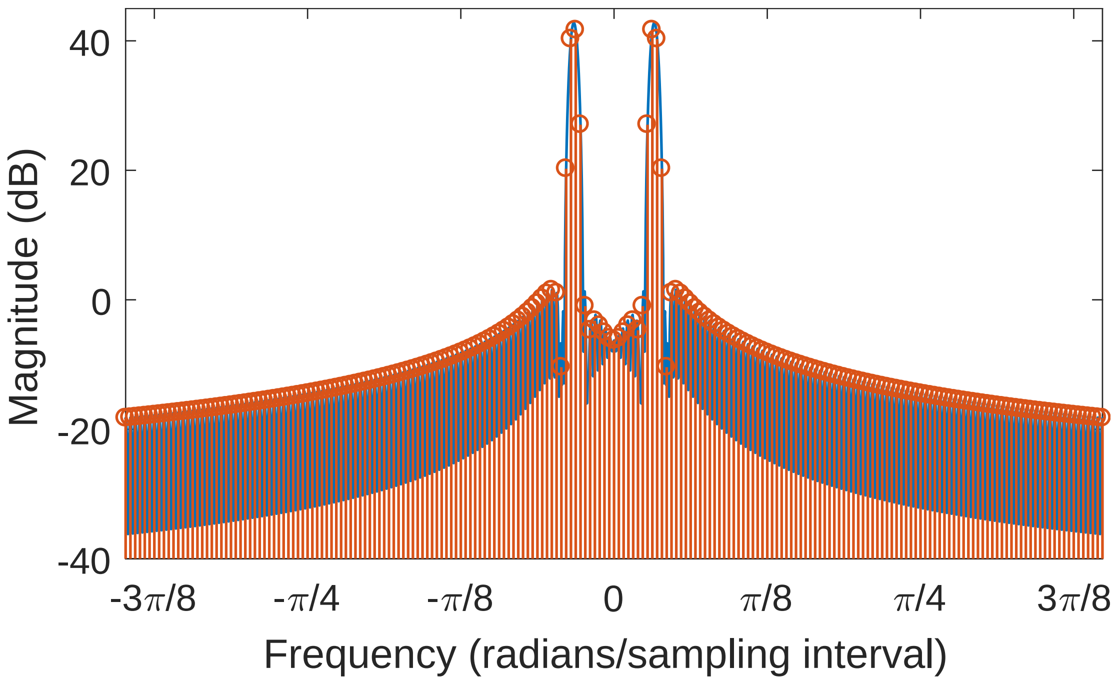
In this case, after the initial cluster of large samples, the leakage near to the frequency of the input signal is lower than in the Hann case, however for the frequencies further away from the peak, the magnitude of the frequency samples are larger than the result of the Hann window, but smaller than when using no (i.e. a rectangular) window.

1.1 Zero padding

The examples shown have selected the minimum length of the DFT, such that $N = L$, however it is possible to increase N, leaving the number of input samples the same, as well as the length of the window. This is termed zero padding as the missing samples from L to $N - 1$ are assumed to be 0. Increasing N by a factor of four, using Hamming window, produces the following result:



It is clear that the shape of the transform has not changed, but that more samples are present in the frequency domain. If we also increase L by a factor of 4, with $N = L$, then more samples of the input signal are being used in the transform. This results in:



In this situation, the window shape is four times narrower because the number of data, and window, samples has been increased by a factor of four.

Some terminology is used in relation to the DFT, such as:

Bin A bin is a term for a frequency domain sample

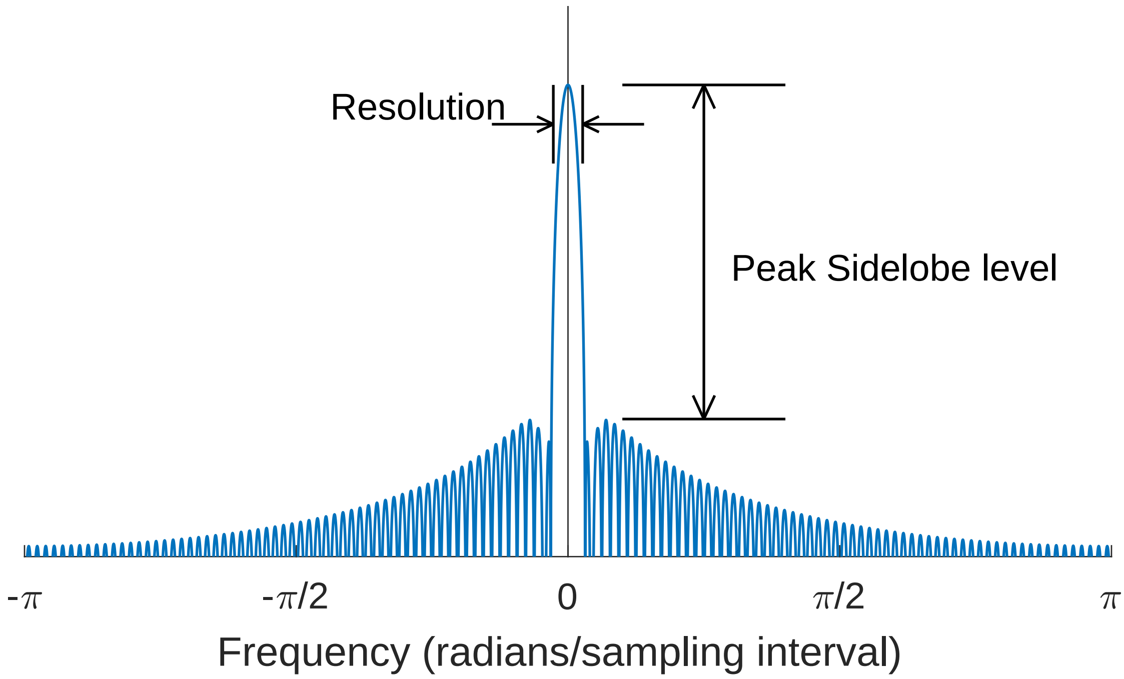
FFT A fast Fourier transform is an efficient method of computing a DFT, provided that N is a power of 2

Main lobe The central lobe, in the frequency domain, of the window. The width of the main lobe is inversely related to the height of the sidelobes

Window	Resolution, -3dB bandwidth (Hz)	Equivalent Noise Bandwidth (ENB) (Hz)	Processing loss (dB)	Peak sidelobe level (dB)	Asymptotic roll-off (dB/octave)
Rectangular	$\frac{0.89}{L\Delta t}$	$\frac{1}{L\Delta t}$	0	-13	-6
Hann	$\frac{1.4}{L\Delta t}$	$\frac{1.5}{L\Delta t}$	4	-32	-18
Hamming	$\frac{1.3}{L\Delta t}$	$\frac{1.36}{L\Delta t}$	2.7	-43	-6
Dolph-Chebyshev	$\frac{1.44}{L\Delta t}$	$\frac{1.51}{L\Delta t}$	3.2	-60	0
Blackman	$\frac{1.52}{L\Delta t}$	$\frac{1.57}{L\Delta t}$	3.4	-51	-6
Kaiser-Bessel	$\frac{1.57}{L\Delta t}$	$\frac{1.65}{L\Delta t}$	3.5	-57	-6

Resolution The minimum frequency separation between two frequencies contained in the input for which the transform can distinguish between them.

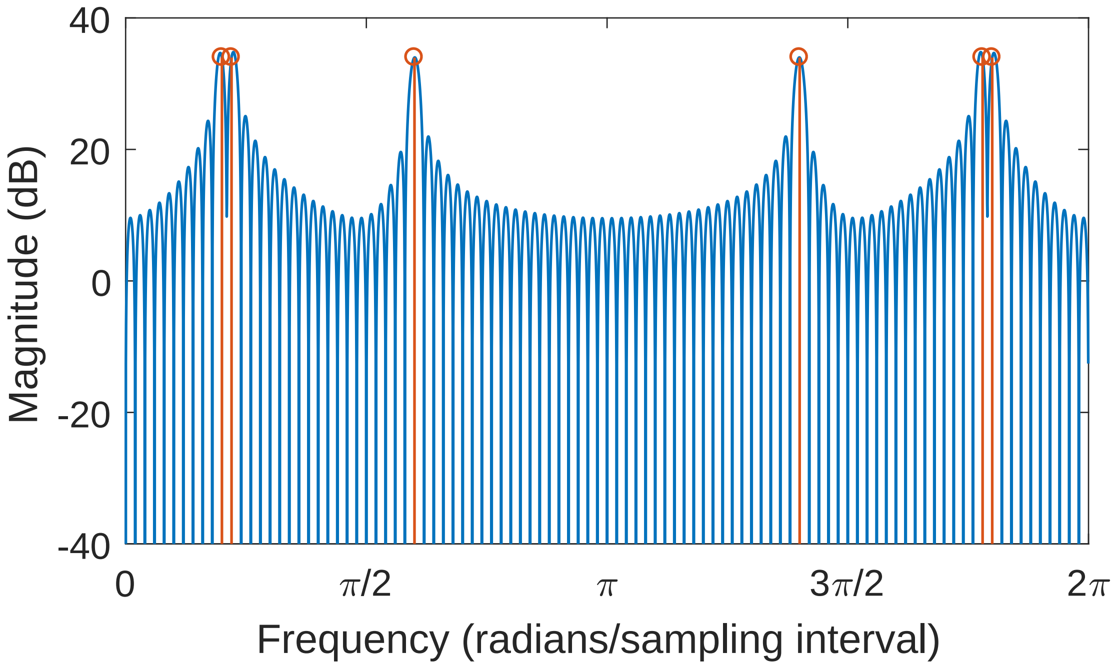
Sidelobes The lobes of the transform that are not the main lobe.



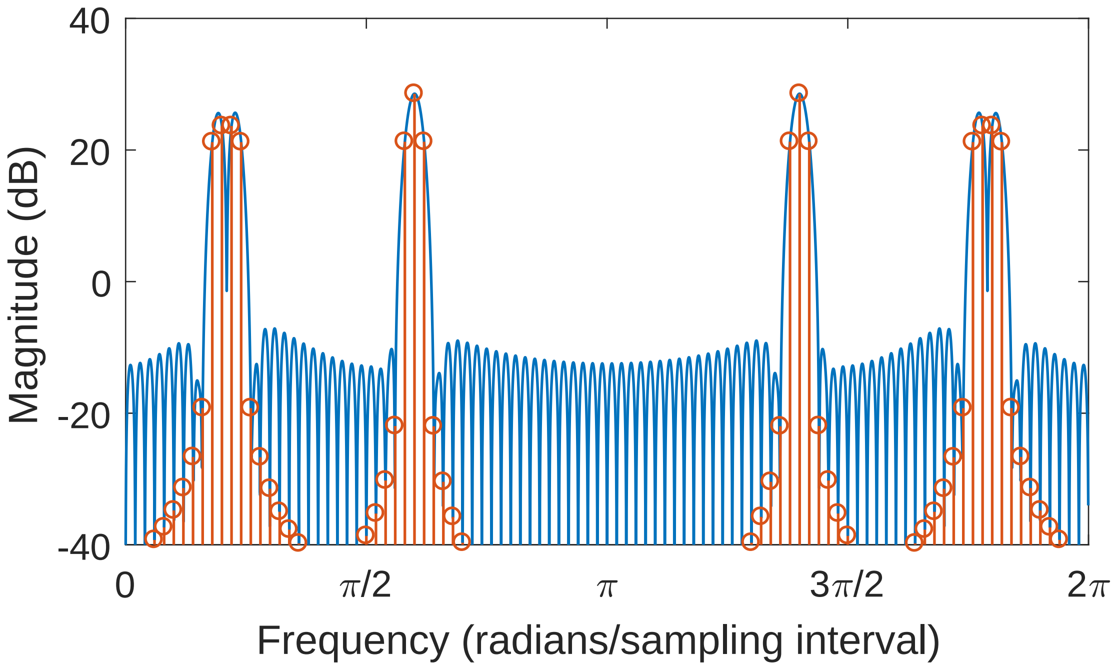
The two features of most concern when selecting a window are generally the width of the main lobe, and the height of the sidelobes. The wider the main lobe is, the poorer the resolution of the transform will be. However, the height of the sidelobes should be kept sufficiently low to allow mixtures of large and small signals to be analysed. This is the dynamic range of the transform. Consider two examples:

$$x(n) = \cos(\omega_0 n) + \cos(\omega_1 n) + \cos(\omega_2 n) \tag{4.8}$$

with $N = L = 100$, $\omega_0 = 0.2\pi$, $\omega_1 = 0.22\pi$ and $\omega_2 = 0.6\pi$. Without using any shaped window, i.e. using only the rectangular window implied by selecting $L = 100$ samples, the transform results in:



Here the first null of $\cos(\omega_0 n)$ lines up with the peak of $\cos(\omega_1 n)$, which is the minimum spacing possible in order to identify the two separate components. When applying the Hamming window the following is obtained:

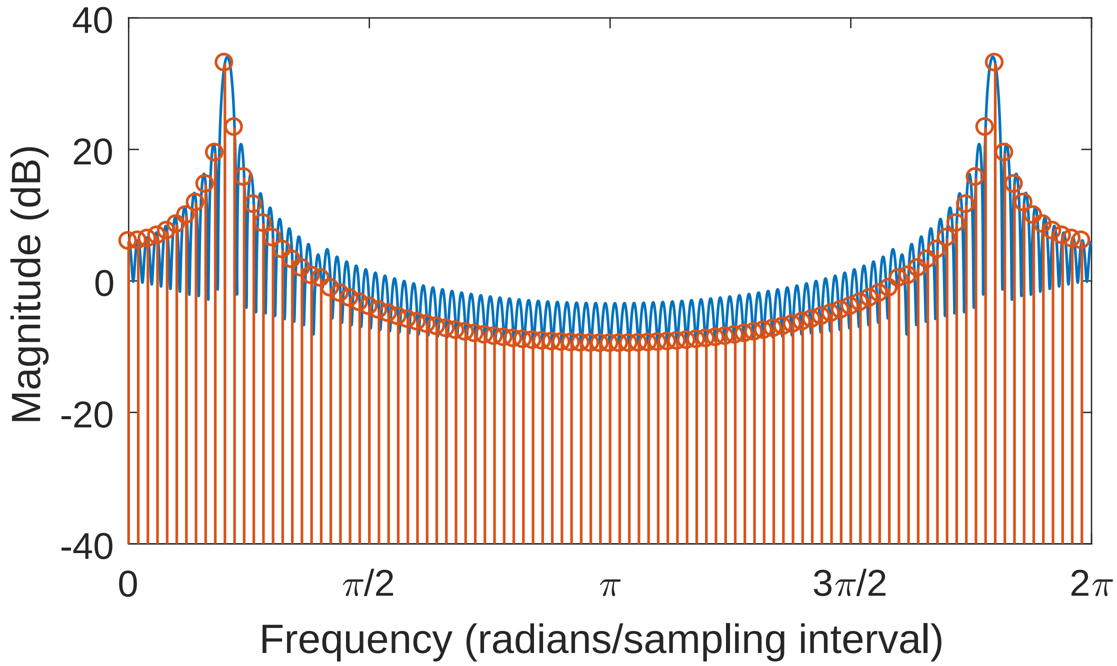


It is clear that the wider main lobe of the Hamming window affects the resolution such that the two frequencies ω_0 and ω_1 can no longer be distinguished. Using this window, more input samples would be required to improve the resolution to be able to once again distinguish the two frequency components.

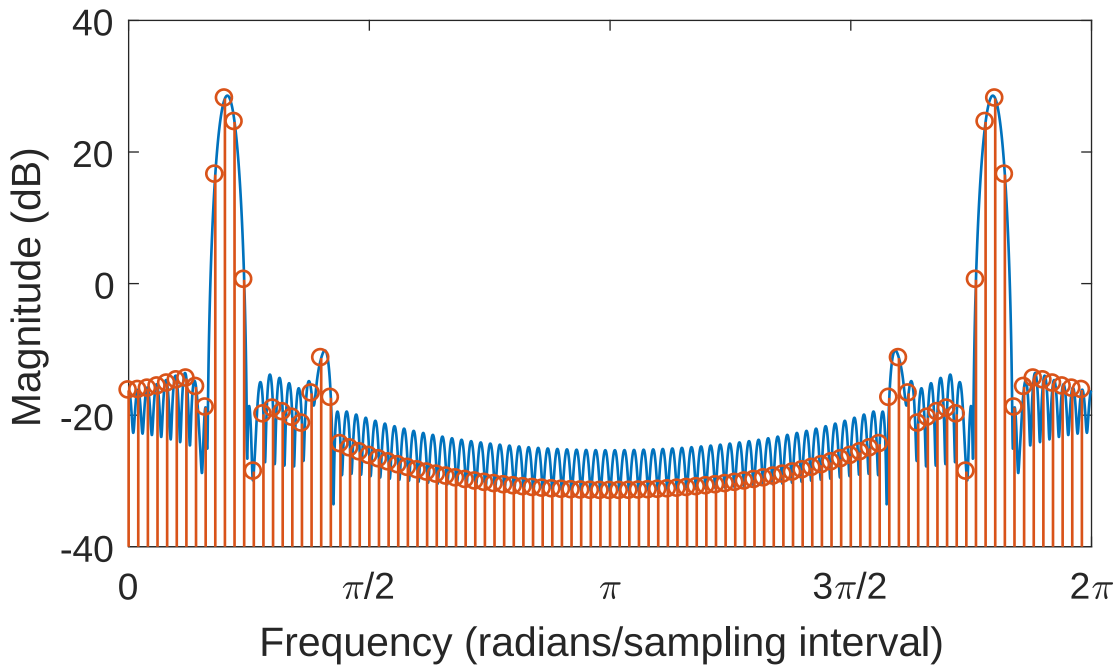
For the second example,

$$x(n) = \cos(\omega_0 n) + 0.01\cos(\omega_1 n)$$

with $N = L = 100$, $\omega_0 = 0.205\pi$ and $\omega_1 = 0.4\pi$.



Here the leakage caused by $\cos(\omega_0 n)$ is so large, that the frequency content of $\cos(\omega_1 n)$ is masked by this. When applying the Hamming window the following is obtained:



It is evident that $\cos(\omega_1 n)$ can now be easily identified.

Other details of concern include the Processing Loss, which describes the effective signal to noise ratio reduction resulting from using the window. For details of this, and other parameters, refer to "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform" by Frederic J. Harris, Proceedings of the IEEE, Vol. 66, No. 1, January 1978, pp. 51-83.