The Discrete Fourier Transform

The discrete-time Fourier transform (DTFT) of a sequence is a continuous function of ω, and repeats with period 2π. In practice we usually want to obtain the Fourier components using digital computation, and can only evaluate them for a discrete set of frequencies. The discrete Fourier transform (DFT) provides a means for achieving this.

The DFT is itself a sequence, and it corresponds roughly to samples, equally spaced in frequency, of the Fourier transform of the signal. The discrete Fourier transform of a length N signal $x[n]$, $n = 0, 1, \ldots, N-1$ is given by

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn}.$$

This is the analysis equation. The corresponding synthesis equation is

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j(2\pi/N)kn}.$$

When dealing with the DFT, it is common to define the complex quantity

$$W_N = e^{-j(2\pi/N)}.$$

With this notation the DFT analysis-synthesis pair becomes

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}.$$

An important property of the DFT is that it is cyclic, with period N, both in the discrete-time and discrete-frequency domains. For example, for any integer r,

$$X[k+rN] = \sum_{n=0}^{N-1} x[n] W_N^{(k+rN)n} = \sum_{n=0}^{N-1} x[n] W_N^{kn} (W_N^N)^{rn}$$
$$= \sum_{n=0}^{N-1} x[n] W_N^{kn} = X[k],$$

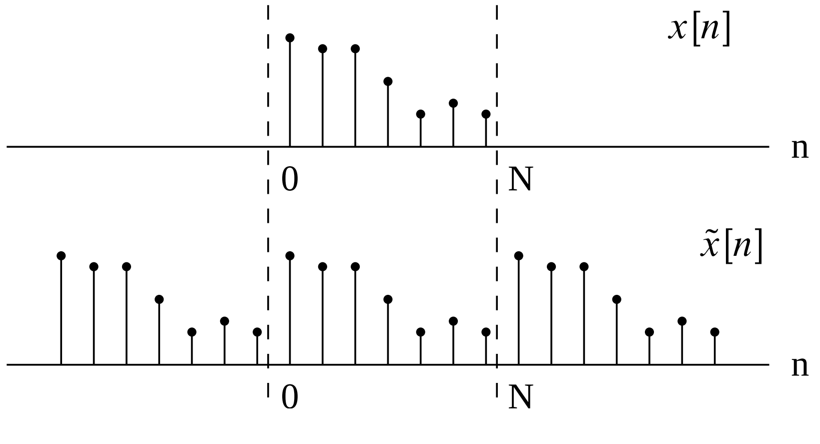
since $W_N^N = e^{-j(2\pi/N)N} = e^{-j2\pi} = 1$. Similarly, it is easy to show that $x[n+rN] = x[n]$, implying periodicity of the synthesis equation. This is important — even though the DFT only depends on samples in the interval 0 to $N-1$, it is implicitly assumed that the signals repeat with period N in both the time and frequency domains.

To this end, it is sometimes useful to define the periodic extension of the signal $x[n]$ to be

$$\tilde{x}[n] = x[n \bmod N] = x[((n))_N].$$

Here $n \bmod N$ and $((n))_N$ are taken to mean n modulo N, which has the value of the remainder after n is divided by N. Alternatively, if n is written in the form $n = kN + l$ for $0 \leq l < N$, then

$$n \bmod N = ((n))_N = l.$$



Similarly, the periodic extension of $X[k]$ is defined to be

$$\tilde{X}[k] = X[k \bmod N] = X[((k))_N].$$

It is sometimes better to reason in terms of these periodic extensions when dealing with the DFT. Specifically, if $X[k]$ is the DFT of $x[n]$, then the inverse DFT of $X[k]$ is $\tilde{x}[n]$. The signals $x[n]$ and $\tilde{x}[n]$ are identical over the interval 0 to $N-1$, but may differ outside of this range. Similar statements can be made regarding the transform $X[k]$.

1 Properties of the DFT

Many of the properties of the DFT are analogous to those of the discrete-time Fourier transform, with the notable exception that all shifts involved must be considered to be circular, or modulo N.

Defining the DFT pairs $x[n] \overset{\mathcal{D}}{\longleftrightarrow} X[k]$, $x_1[n] \overset{\mathcal{D}}{\longleftrightarrow} X_1[k]$, and $x_2[n] \overset{\mathcal{D}}{\longleftrightarrow} X[k]$, the following are properties of the DFT:

- **Symmetry:**

$$
\begin{aligned}
X[k] &= X^*[((-k))_N] \\
\mathrm{Re}\{X[k]\} &= \mathrm{Re}\{X[((-k))_N]\} \\
\mathrm{Im}\{X[k]\} &= -\mathrm{Im}\{X[((-k))_N]\} \\
|X[k]| &= |X[((-k))_N]| \\
\sphericalangle X[k] &= -\sphericalangle X[((-k))_N]
\end{aligned}
$$

- **Linearity:** $ax_1[n] + bx_2[n] \overset{\mathcal{D}}{\longleftrightarrow} aX_1[k] + bX_2[k]$.
- **Circular time shift:** $x[((n-m))_N] \overset{\mathcal{D}}{\longleftrightarrow} W_N^{km} X[k]$.
- **Circular convolution:**

$$\sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N] \overset{\mathcal{D}}{\longleftrightarrow} X_1[k] X_2[k].$$

  Circular convolution between two N-point signals is sometimes denoted by $x_1[n] \; \textcircled{\scriptsize N} \; x[n]$.

- **Modulation:**

$$x_1[n]x_2[n] \stackrel{\mathcal{D}}{\longleftrightarrow} \frac{1}{N}\sum_{l=0}^{N-1} X_1[l]X_2[((k-l))_N].$$

Some of these properties, such as linearity, are easy to prove. The properties involving time shifts can be quite confusing notationally, but are otherwise quite simple. For example, consider the 4-point DFT

$$X[k] = \sum_{n=0}^{3} x[n]W_4^{kn}$$

of the length 4 signal $x[n]$. This can be written as

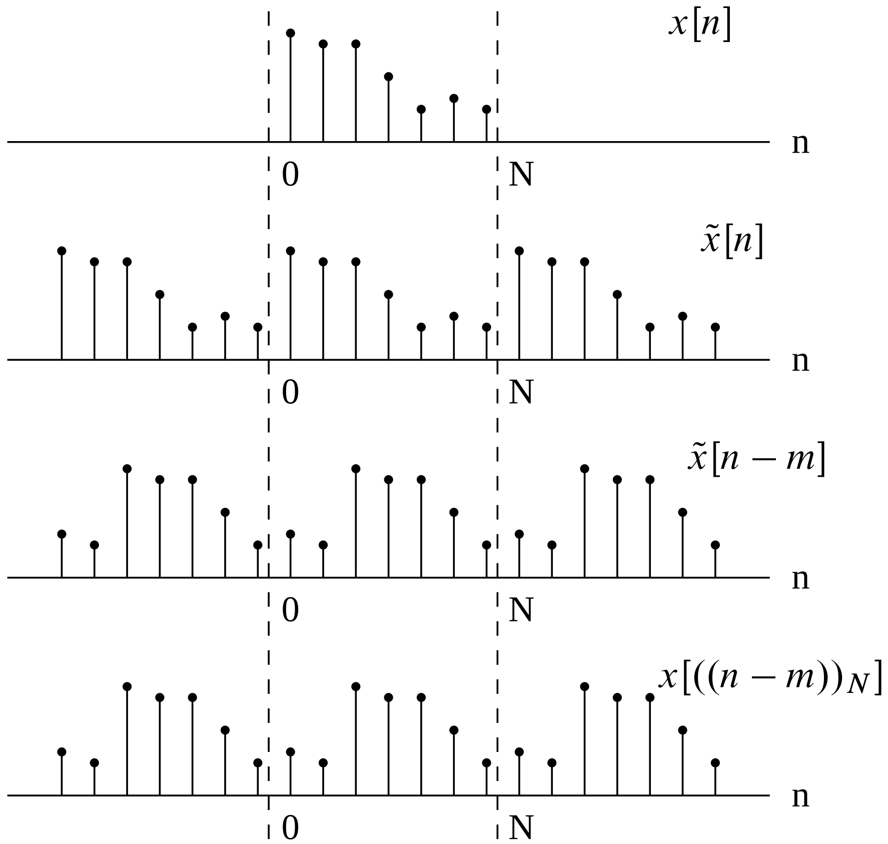
$$X[k] = x[0]W_4^{0k} + x[1]W_4^{1k} + x[2]W_4^{2k} + x[3]W_4^{3k}$$

The product $W_4^{1k}X[k]$ can therefore be written as

$$\begin{aligned} W_4^{1k}X[k] &= x[0]W_4^{1k} + x[1]W_4^{2k} + x[2]W_4^{3k} + x[3]W_4^{4k} \\ &= x[3]W_4^{0k} + x[0]W_4^{1k} + x[1]W_4^{2k} + x[2]W_4^{3k} \end{aligned}$$

since $W_4^{4k} = W_4^{0k}$. This can be seen to be the DFT of the sequence $x[3], x[0], x[1], x[2]$, which is precisely the sequence $x[n]$ *circularly shifted* to the right by one sample. This proves the time-shift property for a shift of length 1. In general, multiplying the DFT of a sequence by W_N^{km} results in an N-point *circular* shift of the sequence by m samples. The convolution properties can be similarly demonstrated.

It is useful to note that the circularly shifted signal $x[((n-m))_N]$ is the same as the linearly shifted signal $\tilde{x}[n-m]$, where $\tilde{x}[n]$ is the N-point periodic extension of $x[n]$.



On the interval 0 to $N-1$, the circular convolution

$$x_3[n] = x_1[n] \circledN x_2[n] = \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$$

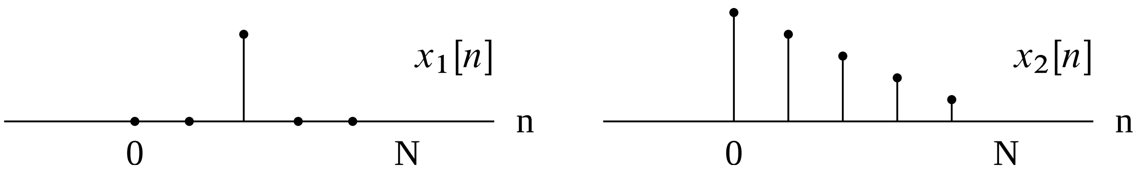
can therefore be calculated using the linear convolution product

$$x_3[n] = \sum_{m=0}^{N-1} x_1[m] \tilde{x}_2[n-m].$$

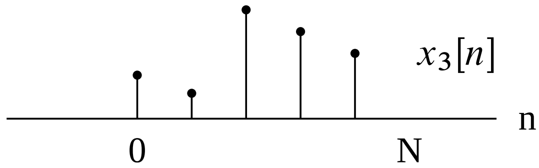
Circular convolution is really just periodic convolution.

Example: Circular convolution with a delayed impulse sequence

Given the sequences



the circular convolution $x_3[n] = x_1[n] \circledN x_2[n]$ is the signal $\tilde{x}[n]$ delayed by two samples, evaluated over the range 0 to $N-1$:



Example: Circular convolution of two rectangular pulses

Let

$$x_1[n] = x_2[n] = \begin{cases} 1 & 0 \le n \le L-1 \\ 0 & \text{otherwise.} \end{cases}$$

If $N = L$, then the N-point DFTs are

$$X_1[k] = X_2[k] = \sum_{n=0}^{N-1} W_N^{kn} = \begin{cases} N & k = 0 \\ 0 & \text{otherwise.} \end{cases}$$

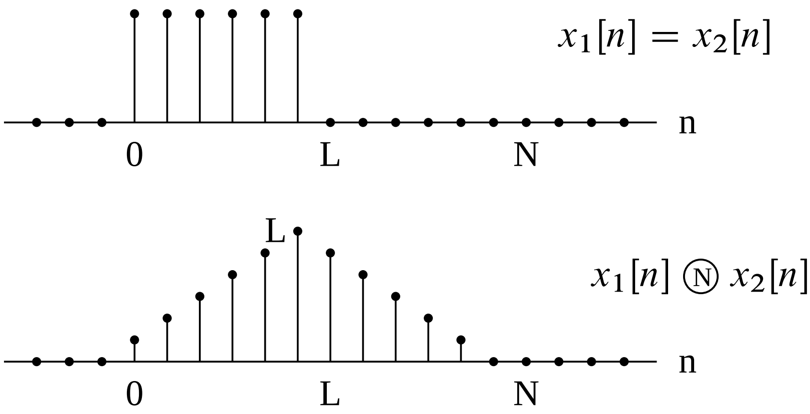
Since the product is

$$X_3[k] = X_1[k]X_2[k] = \begin{cases} N^2 & k = 0 \\ 0 & \text{otherwise,} \end{cases}$$

it follows that the N-point circular convolution of $x_1[n]$ and $x_2[n]$ is

$$x_3[n] = x_1[n] \circledN x_2[n] = N, \qquad 0 \le n \le N-1.$$

Suppose now that $x_1[n]$ and $x_2[n]$ are considered to be length $2L$ sequences by augmenting with zeros. The $N = 2L$-point circular convolution is then seen to be the same as the linear convolution of the finite-duration sequences $x_1[n]$ and $x_2[n]$:



2 Linear convolution using the DFT

Using the DFT we can compute the circular convolution as follows

- Compute the N-point DFTs $X_1[k]$ and $X_2[k]$ of the two sequences $x_1[n]$ and $x_2[n]$.
- Compute the product $X_3[k] = X_1[k]X_2[k]$ for $0 \leq k \leq N-1$.
- Compute the sequence $x_3[n] = x_1[n] \circledN x_2[n]$ as the inverse DFT of $X_3[k]$.

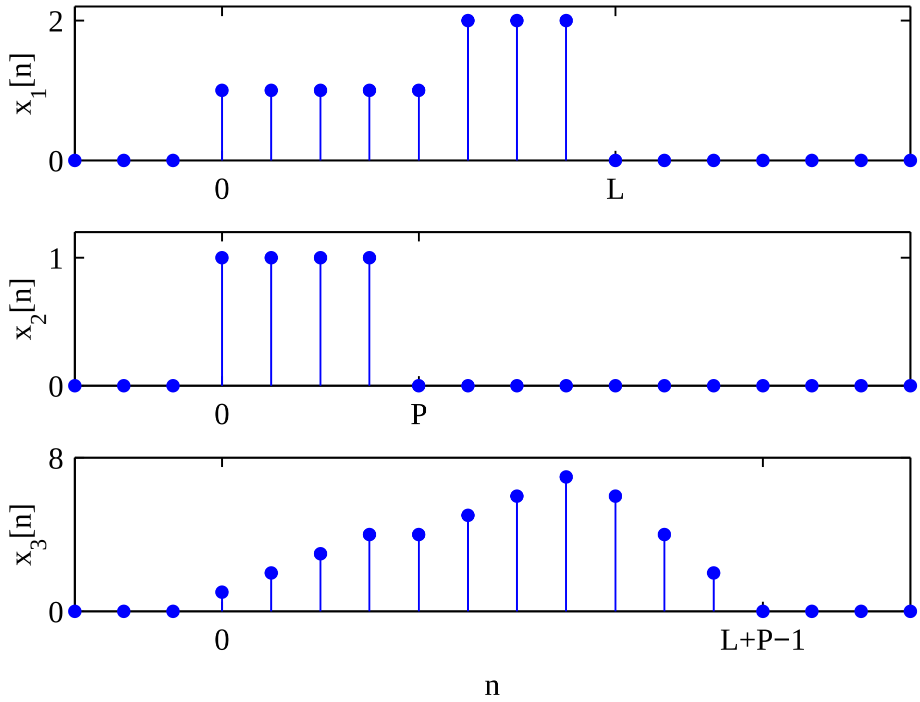
This is computationally useful due to efficient algorithms for calculating the DFT. The question that now arises is this: how do we get the *linear* convolution (required in speech, radar, sonar, image processing) from this procedure?

2.1 Linear convolution of two finite-length sequences

Consider a sequence $x_1[n]$ with length L points, and $x_2[n]$ with length P points. The linear convolution of the sequences,

$$x_3[n] = \sum_{m=-\infty}^{\infty} x_1[m]x_2[n-m],$$

is nonzero over a maximum length of $L + P - 1$ points:



Therefore $L + P - 1$ is the maximum length of $x_3[n]$ resulting from the linear convolution.

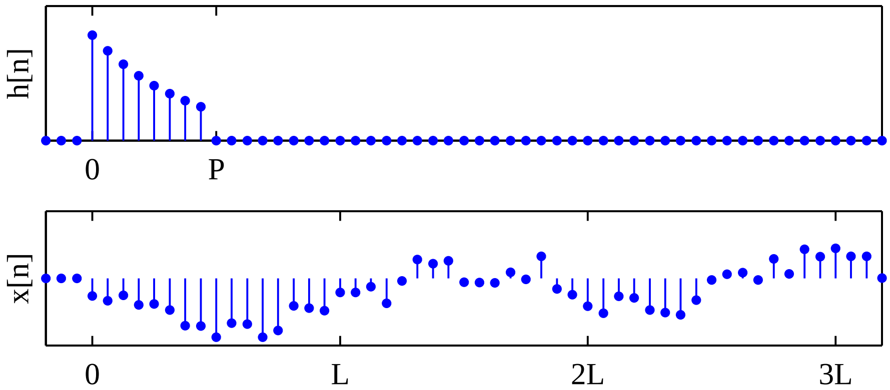
The N-point circular convolution of $x_1[n]$ and $x_2[n]$ is

$$x_1[n] \circledN x_2[n] = \sum_{m=0}^{N-1} x_1[m]x_2[((n-m))_N] = \sum_{m=0}^{N-1} x_1[m]\tilde{x}_2[n-m] :$$

It is easy to see that the circular convolution product will be equal to the linear convolution product on the interval 0 to $N - 1$ as long as we choose $N \geq L + P - 1$. The process of augmenting a sequence with zeros to make it of a required length is called **zero padding**.

2.2 Convolution by sectioning

Suppose that for computational efficiency we want to implement a FIR system using DFTs. It cannot in general be assumed that the input signal has a finite duration, so the methods described up to now cannot be applied directly:

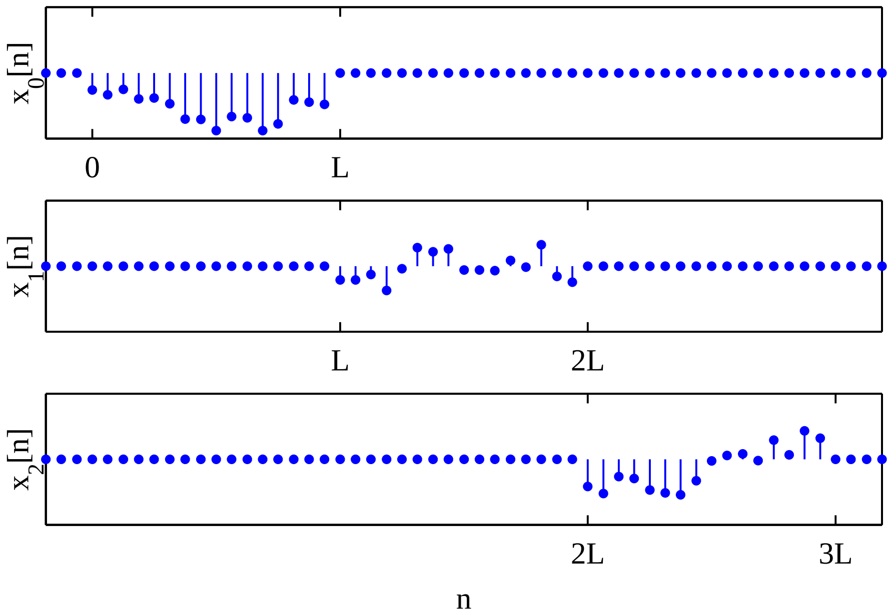


The solution is to use **block convolution**, where the signal to be filtered is segmented into sections of length L. The input signal $x[n]$, here assumed to be causal, can be decomposed into blocks of length L as follows:

$$x[n] = \sum_{r=0}^{\infty} x_r[n - rL],$$

where

$$x_r[n] = \begin{cases} x[n + rL] & 0 \leq n \leq L - 1 \\ 0 & \text{otherwise.} \end{cases}$$

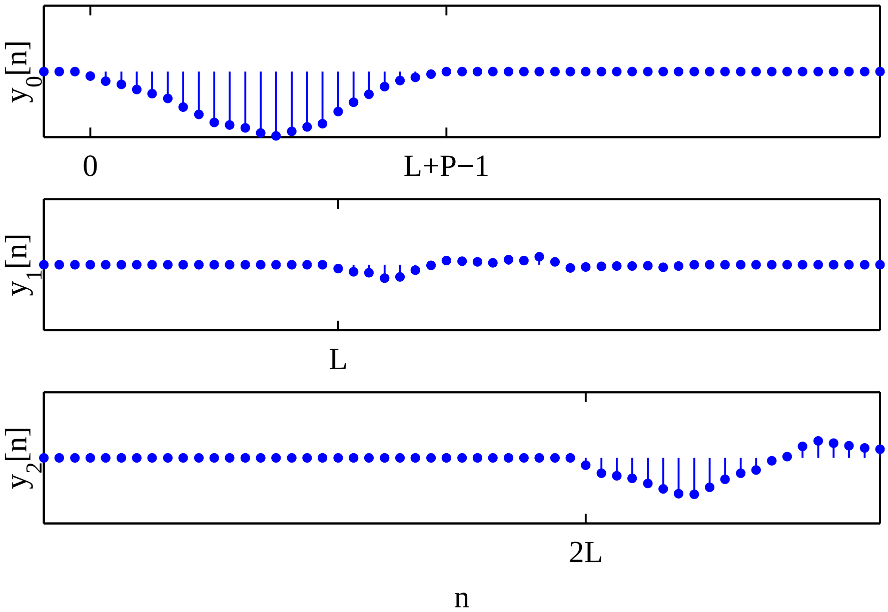


The convolution product can therefore be written as

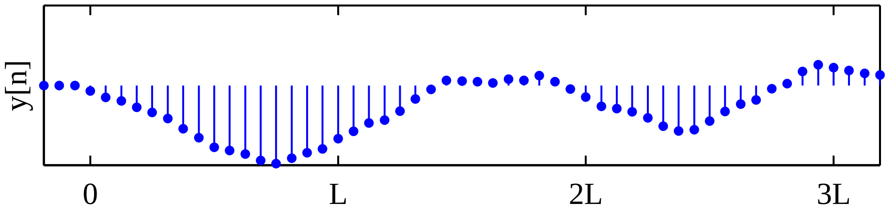
$$y[n] = x[n] * h[n] = \sum_{r=0}^{\infty} y_r[n - rL],$$

where $y_r[n]$ is the response

$$y_r[n] = x_r[n] * h[n].$$



Since the sequences $x_r[n]$ have only L nonzero points and $h[n]$ is of length P, each response term $y_r[n]$ has length $L + P - 1$. Thus linear convolution can be obtained using N-point DFTs with $N \geq L + P - 1$. Since the final result is obtained by summing the overlapping output regions, this is called the **overlap-add** method.



An alternative block convolution procedure, called the **overlap-save** method, corresponds to implementing an L-point circular convolution of a P-point impulse response $h[n]$ with an L-point segment $x_r[n]$. The portion of the output that corresponds to linear convolution is then identified (consisting of $L - (P - 1)$ points), and the resulting segments patched together to form the output.

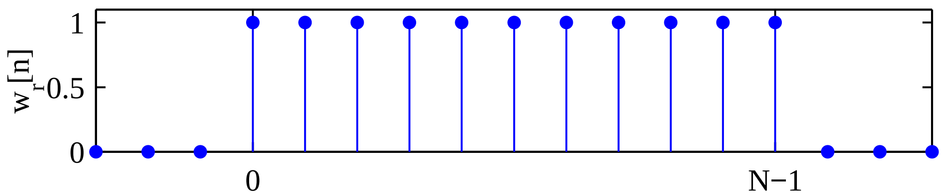
3 Spectrum estimation using the DFT

Spectrum estimation is the task of estimating the DTFT of a signal $x[n]$. The DTFT of a discrete-time signal $x[n]$ is

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

The signal $x[n]$ is generally of infinite duration, and $X(e^{j\omega})$ is a continuous function of ω. The DTFT can therefore not be calculated using a computer.

Consider now that we truncate the signal $x[n]$ by multiplying with the rectangular window $w_r[n]$:



The windowed signal is then $x_w[n] = x[n]w_r[n]$. The DTFT of this windowed signal is given by

$$X_w(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_w[n]e^{-j\omega n} = \sum_{n=0}^{N-1} x_w[n]e^{-j\omega n}.$$

Noting that the DFT of $x_w[n]$ is

$$X_w[k] = \sum_{n=0}^{N-1} x_w[n]e^{-j\frac{2\pi kn}{N}},$$

it is evident that

$$X_w[k] = X_w(e^{j\omega})\big|_{\omega=2\pi k/N}.$$

The values of the DFT $X_w[k]$ of the signal $x_w[n]$ are therefore periodic samples of the DTFT $X_w(e^{j\omega})$, where the spacing between the samples is $2\pi/N$. Since the relationship between the discrete-time frequency variable and the continuous-time frequency variable is $\omega = \Omega T$, the DFT frequencies correspond to continuous-time frequencies

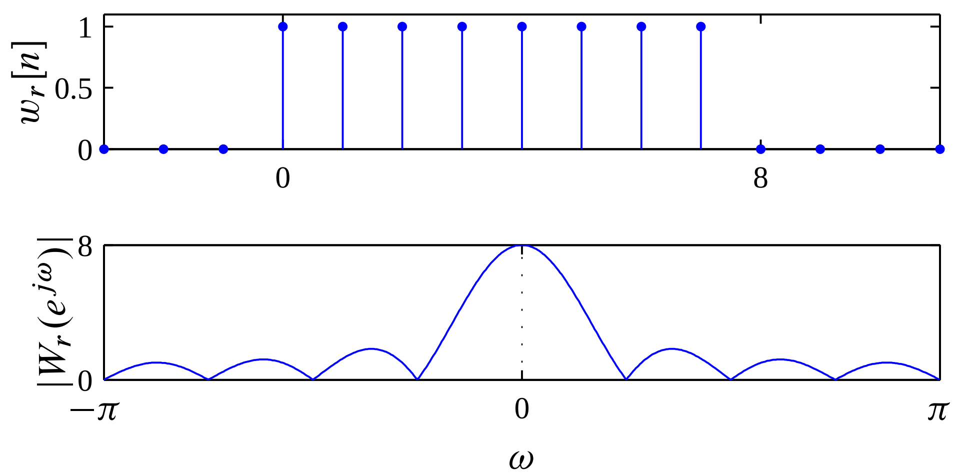
$$\Omega_k = \frac{2\pi k}{NT}.$$

The DFT can therefore only be used to find points on the DTFT of the *windowed* signal $x_w[n]$ of $x[n]$.

The operation of windowing involves multiplication in the discrete time domain, which corresponds to continuous-time periodic convolution in the DTFT frequency domain. The DTFT of the windowed signal is therefore

$$X_w(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})W(e^{j(\omega-\theta)})d\theta,$$

where $W(e^{j\omega})$ is the frequency response of the window function. For a simple rectangular window, the frequency response is as follows:

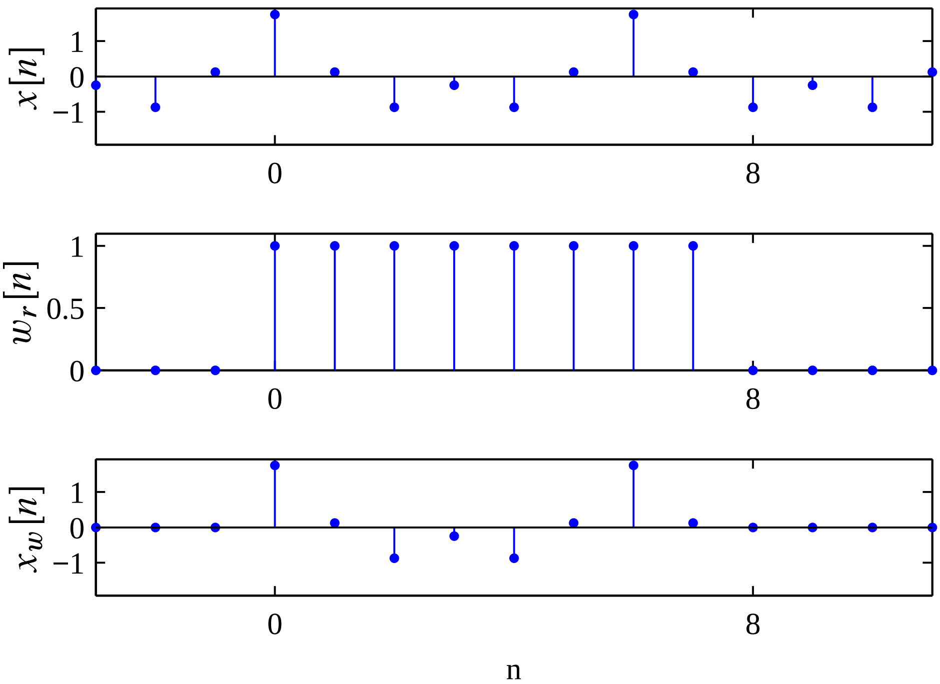


The DFT therefore effectively samples the DTFT of the signal convolved with the frequency response of the window.

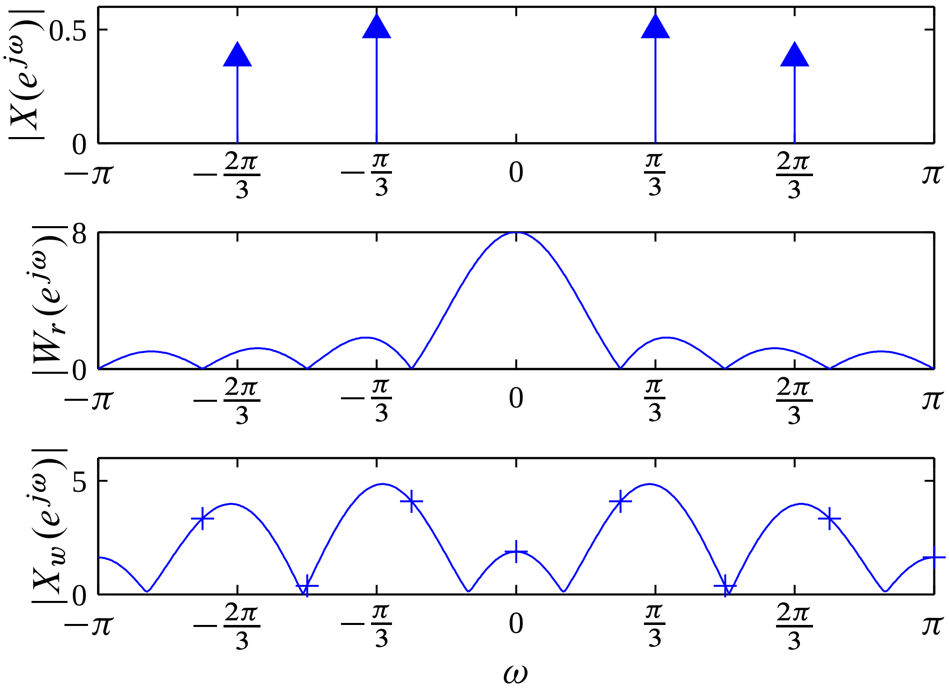
Example: Spectrum analysis of sinusoidal signals Suppose we have the sinusoidal signal combination

$$x[n] = \cos(\pi/3n) + 0.75\cos(2\pi/3n), \qquad -\infty < n < \infty.$$

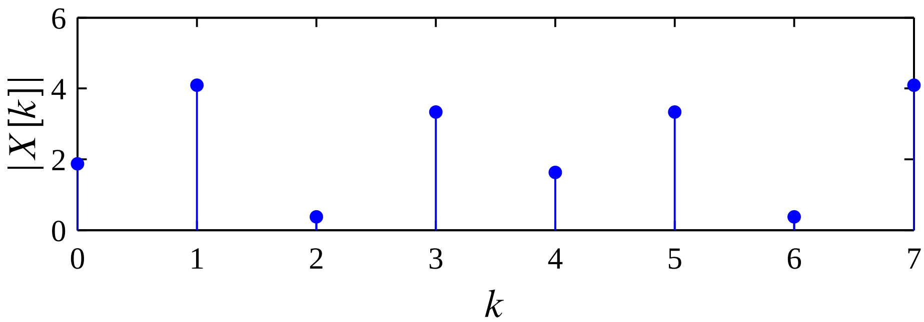
Since the signal is infinite in duration, the DTFT cannot be computed numerically. We therefore window the signal in order to make the duration finite:



The operation of windowing modifies the signal. This is reflected in the discrete-time Fourier transform domain by a *spreading* of the frequency components:



The operation of windowing therefore limits the ability of the Fourier transform to resolve closely-spaced frequency components. When the DFT is used for spectrum estimation, it effectively samples the spectrum of this modified signal at the locations of the crosses indicated:



Note that since $k = 0$ corresponds to $\omega = 0$, there is a corresponding shift in the sampled values.

In general, the elements of the N-point DFT of $x_w[n]$ contain N evenly-spaced samples from the DTFT $X_w(e^{j\omega})$. These samples span an entire period of the DTFT, and therefore correspond to frequencies at spacings of $2\pi/N$. We can obtain samples with a closer spacing by performing more computation.

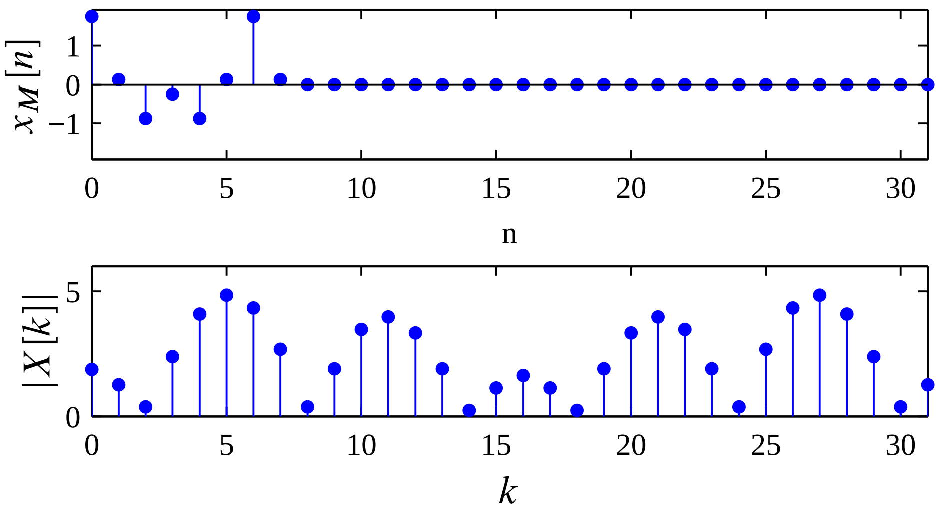
Suppose we form the zero-padded length M signal $x_M[n]$ as follows:

$$x_M[n] = \begin{cases} x[n] & 0 \leq n \leq N-1 \\ 0 & N \leq n \leq M-1. \end{cases}$$

The M-point DFT of this signal is

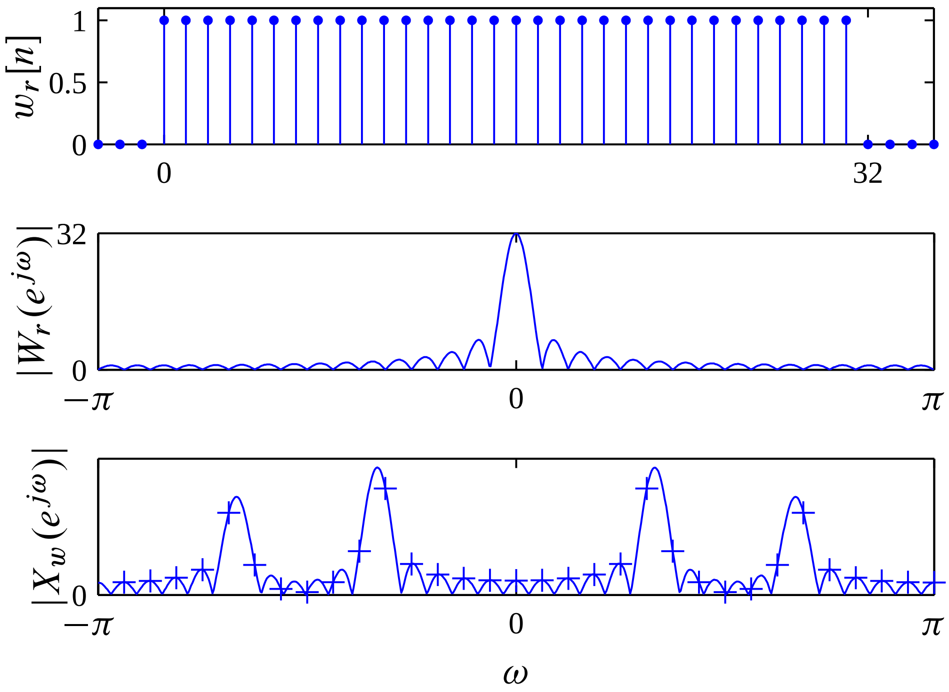
$$X_M[k] = \sum_{n=0}^{M-1} x_M[n] e^{-j\frac{2\pi}{M}kn} = \sum_{n=0}^{N-1} x_w[n] e^{-j\frac{2\pi}{M}kn}$$
$$= \sum_{n=-\infty}^{\infty} x_w[n] e^{-j\frac{2\pi}{M}kn}$$

The sample $X_p[k]$ can therefore be seen to correspond to the DTFT of the the windowed signal $x_w[n]$ at frequency $\omega_k = 2\pi k/M$. Since M is chosen to be larger than N, the transform values correspond to regular samples of $X_w(e^{j\omega})$ with a closer spacing of $2\pi/M$. The following figure shows the magnitude of the DFT transform values for the 8-point signal shown previously, but zero-padded to use a 32-point DFT:

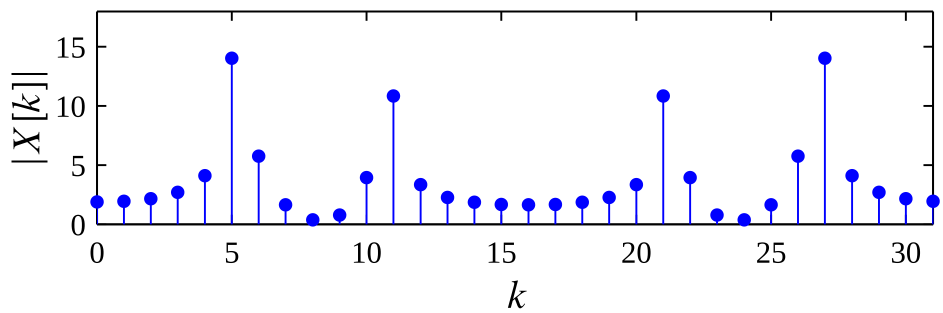


Note that this process increases the density of the samples, but has no effect on the resolution of the spectrum.

If $W(e^{j\omega})$ is sharply peaked, and approximates a Dirac delta function at the origin, then $X_w(e^{j\omega}) \approx X(e^{j\omega})$. The values of the DFT then correspond quite accurately to samples of the DTFT of $x[n]$. For a rectangular window, the approximation improves as N increases:

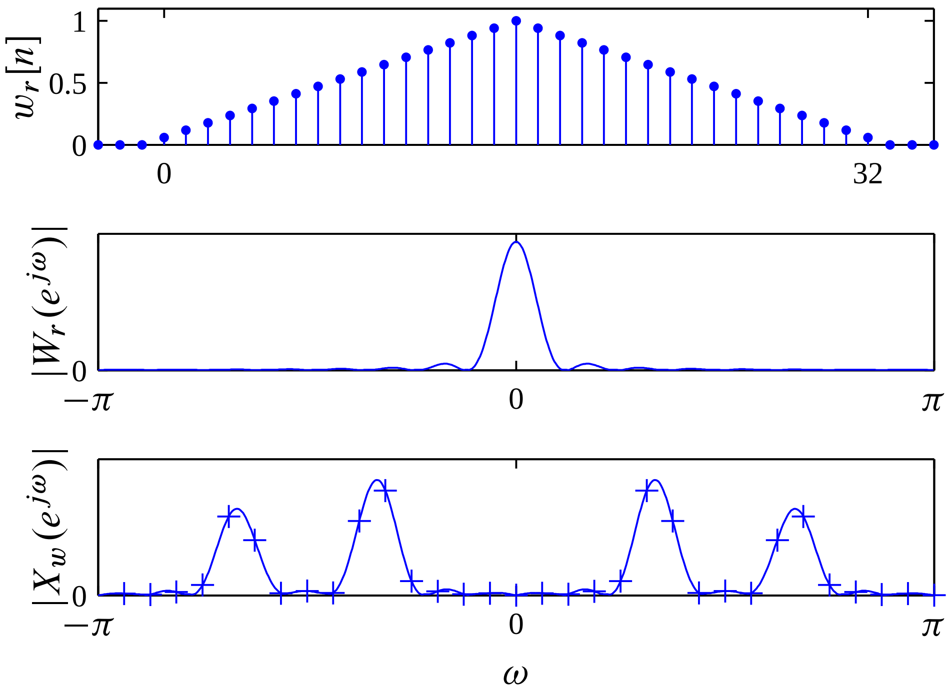


The magnitude of the DFT of the windowed signal is

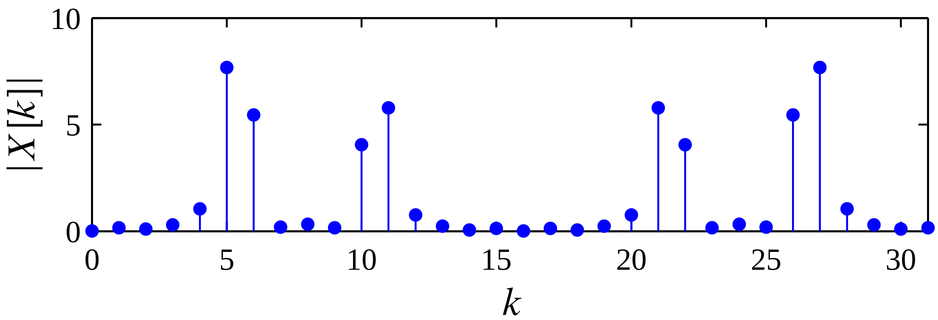


which is clearly easier to interpret than for the case of the shorter signal. As the window length tends to ∞, the relationship becomes exact.

The rectangular window inherent in the DFT has the disadvantage that the peak sidelobe of $W_r(e^{j\omega})$ is high relative to the mainlobe. This limits the ability of the DFT to resolve frequencies. Alternative windows may be used which have preferred behaviour — the only requirement is that in the time domain the window function is of finite duration. For example, the triangular window



leads to DFT samples with magnitude



Here the sidelobes have been reduced at the cost of diminished resolution — the mainlobe has become wider.

The method just described forms the basis for the *periodogram* spectrum estimate. It is often used in practice on account of its perceived simplicity. However, it has a poor statistical properties — *model-based* spectrum estimates generally have higher resolution and more predictable performance.

Finally, note that the discrete samples of the spectrum are only a complete representation if the sampling criterion is met. The samples therefore have to be sufficiently closely spaced.

4 Fast Fourier transforms

The widespread application of the DFT to convolution and spectrum analysis is due to the existence of fast algorithms for its implementation. The class of methods are referred to as **fast Fourier transforms** (FFTs).

Consider a direct implementation of an 8-point DFT:

$$X[k] = \sum_{n=0}^{7} x[n] W_8^{kn}, \qquad k = 0, \ldots, 7.$$

If the factors W_8^{kn} have been calculated in advance (and perhaps stored in a lookup table), then the calculation of $X[k]$ for each value of k requires 8 complex multiplications and 7 complex additions. The 8-point DFT therefore requires 8×8 multiplications and 8×7 additions. For an N-point DFT these become N^2 and $N(N-1)$ respectively. If $N = 1024$, then approximately one million complex multiplications and one million complex additions are required.

The key to reducing the computational complexity lies in the observation that the same values of $x[n]W_8^{kn}$ are effectively calculated many times as the computation proceeds — particularly if the transform is long.

The **conventional decomposition** involves **decimation-in-time**, where at each stage a N-point transform is decomposed into two $N/2$-point transforms. That is, $X[k]$ can be written as

$$X[k] = \sum_{r=0}^{N/2-1} x[2r]W_N^{2rk} + \sum_{r=0}^{N/2-1} x[2r+1]W_N^{(2r+1)k}$$
$$= \sum_{r=0}^{N/2-1} x[2r](W_N^2)^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1](W_N^2)^{rk}.$$

Noting that $W_N^2 = W_{N/2}$ this becomes

$$X[k] = \sum_{r=0}^{N/2-1} x[2r](W_{N/2})^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1](W_{N/2})^{rk}$$
$$= G[k] + W_N^k H[k].$$

The original N-point DFT can therefore be expressed in terms of two $N/2$-point DFTs.

The $N/2$-point transforms can again be decomposed, and the process repeated until only 2-point transforms remain. In general this requires $log_2 N$ stages of decomposition. Since each stage requires approximately N complex multiplications, the complexity of the resulting algorithm is of the order of $N \log_2 N$.

The difference between N^2 and $N \log_2 N$ complex multiplications can become considerable for large values of N. For example, if $N = 2048$ then $N^2/(N \log_2 N) \approx 200$.

There are numerous variations of FFT algorithms, and all exploit the basic redundancy in the computation of the DFT. In almost all cases an off-the-shelf implementation of the FFT will be sufficient — there is seldom any reason to implement a FFT yourself.