

EE4 Tutorial

Convolution

Convolution is the term given to the mathematical technique for determining a system output given an input signal and the system impulse response. It is written as

$$x(t) = \int_0^{\infty} x(t - \tau) h(\tau) d\tau \quad (1)$$

When you start the convolution tool you will find that you can select signals, run animations, and control the display properties. The current input signal and system impulse response are displayed - you can change the data in these graphs by clicking and dragging with the mouse or you can select pre-defined signals using the menus. It is helpful if you ensure that the signal and system response are coloured differently.

The other graphs show the time-reversed input signal and the system response, the product of the time-reversed input signal (due to the $-\tau$ in equation (1)) and the system response, and their convolution. The product and convolution data will change as the time-reversed input signal is moved relative to the system impulse response.

When the system impulse response is a unit pulse and the input is a ramp function, you can test out the convolution process by clicking on the $-\tau$ symbol in the combined time-reversed system impulse response and signal graph and dragging the time reversed input signal backwards and forwards through the system impulse response. You can also let the tool do this for you by clicking on the double arrow button.

As the ramp signal is shifted through the unit pulse, the product graph clearly shows that the convolution is calculating the area under a selected portion of the ramp signal. The convolution graph plots the increasing, and then decreasing area of this product. The system with this impulse response is known as a finite time integrator as it performs integration over a finite length of the input signal. Experiment with other input signals such as sine waves, other unit pulses, impulses, etc.

Convolution with an impulse

Change the system response to a single impulse at 2ms. As you perform the convolution, as before, you will find that the output is a copy of the input signal, except that it is delayed by 2ms. In this instance, the impulse is performing a sampling function of the input signal as described by

$$\int_0^{\infty} x(t - \tau) \delta(\tau - u) d\tau = x(t - u)$$

Examination of the product will reveal that the integrand, $x(t - \tau) \delta(\tau - u)$, is also an impulse. In fact, the impulse is a scaled unit impulse, the scaling resulting in an impulse of area $x(t - u)$, hence the final convolution result of $x(t - u)$. Thus, analytically we find that the system response of a unit impulse

results in the output of the system being a copy of the input signal, with the only difference being a time shift.

As a heuristic argument as to why this should be the case, think of the system response as the system output when the input signal is an impulse function at time $t = 0$. When the system response is also an impulse, then the input signal shape doesn't alter. If that impulse is at a non-zero time value, then there is a time shift between the input and the output of the system. (For real systems any time shift must represent a delay, otherwise the system would predict its own input signal!)

A system that sums two delayed versions of the input signal is represented by two impulses in the system response. Experiment with the effects that you can obtain by examining system responses consisting of impulses.

Discrete Convolution

Convolution, as just described using integrals, has a discrete time parallel which is more appropriate to digital signal processing systems.

For the mathematics, the integration is replaced by a summation, hence convolution may be re-expressed as

$$y[k] = \sum_{n=0}^{\infty} x[k-n] h[n] \quad (2)$$

The same experiments carried out for the continuous system may be repeated for the discrete case, with similar results. Select a system of 16 sample length unit pulse, and a signal of a sampled ramp. Sliding the signal over the system response, the discrete time integrator can be seen to operate. The product of the two waveforms is displayed on the middle graph, and the sum of the product samples displayed on the lower graph.

Unit pulse response

Set the system response to be a unit sample at sample number 8, and select any signal that you desire. As you slide the signal through the system response you will observe that the discrete convolution result is a copy of the signal, delayed by 8 samples. This illustrates the discrete time equivalent system response to the impulse in the system response for the continuous time system. Thus the system response is described as the unit pulse response in a discrete time system, while it is described as the impulse response in a continuous time system.

Again, experiment with different system responses, and signals, designing your own on the signal window if you wish to try out signals that are not pre-defined.

The Application of Discrete Time Convolution

Discrete time convolution can be used to determine the output of a sampled data system from its input and pulse response. A commonly used discrete system is a finite impulse response (FIR) filter.

This tool presents you with a schematic of a FIR Filter. The boxes marked Z^{-1} represent delay elements whose output is the previous value of their input. Using the single step button you can observe the input signal, $x[n]$ as its samples work their way through the delay elements.

There are a number of multiplication symbols in the filter with weights, which are the system response, associated with them. The product of these weights and the input signal samples are summed together to give the output of the filter, $y[n]$

You can view the values at various points in the filter as either a bar denoting the size of the value, by a number, or by both means. The colours of the values or bars correspond to the colours of the graphs. Observing both windows simultaneously, the mechanism by which the convolution is performed can be seen. The system response is coded in the multiplication weights, the product of the input signal and the system response is calculated at the output of the multipliers, and the final sum is calculated by the summation unit driving the filter output.

Experiment with setting various system responses and signals such as setting a single value of the input to 1 to convince yourself that the system response values of the filter define the unit pulse response of the filter.

Correlation

Correlation is a technique that is very similar in mechanism to convolution, with one subtle difference.

The correlation integral is given by

$$R_{yx}(\lambda) = \int_{-\infty}^{\infty} y(t) x(t + \lambda) dt$$

While this may appear to be similar to equation (1) with some changes of variable, unlike convolution where $x(t)$ is time reversed, in correlation this is not the case. The integral thus no longer represents the output of a filter driven by an input signal. Rather it is a tool used to measure the similarity between two signals. A large correlation value (positive or negative) represents a strong similarity between the two signals, while a value near zero represents little similarity. The delay value, λ allows this comparison to be made on two signals at different delay separations.

Using the correlation tool compare a number of signals with a unit pulse shape, noting the maximum values that you can obtain, and the delay at which these values occur. For all signals that you correlate, except for those signals incorporating impulses, the maximum value that you will obtain will be found by those signals similar in shape to the one you are correlating with. This is because correlation identifies the similarity between the two signals.

If a signal is correlated with itself, a process known as autocorrelation, then the maximum value of the correlation can be found at a time shift of 0. Verify this yourself for any signals that you wish to test out. For the correlation between two different signals, known as cross-correlation, this is not necessarily the case.

Fourier Transforms

The Fourier transform defines a relationship between a signal in the time domain and its representation in the frequency domain. Being a transform, no information is created or lost in the process, so the original signal can be recovered from knowing the Fourier transform, and vice versa.

The Fourier transform of a signal is a continuous complex valued signal capable of representing real valued or complex valued continuous time signals.

The tool allows you to view these complex valued signals as either their real and quadrature (also known as imaginary) components separately, or by a magnitude and phase representation. You may switch between these two representations at any point. Mathematically switching between the two representations for a given complex value can be expressed as

$$|X| = \sqrt{X_r^2 + X_i^2} \quad \text{and} \quad \angle X = \tan^{-1}\left(\frac{X_i}{X_r}\right)$$

or equivalently,

$$X_r = |X| \cos(\angle X) \quad \text{and} \quad X_i = |X| \sin(\angle X)$$

where $|X|$ and $\angle X$ are the magnitude and phase of the complex number, and X_r and X_i are the real and quadrature components of the complex number. In this tool, the magnitude is plotted on a dB scale. Select a few signals, such as unit pulses and sine waves, and view them using the two methods to see how they are related.

The Fourier transform itself is defined by the equation

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt$$

where $X(f)$ is the Fourier transform of $x(t)$. Frequency is measured in Hertz, with f as the frequency variable.

Fourier transform of signals

Using the tool, display the Fourier transform of a 4ms unit pulse. You will observe that the frequency response is a continuous signal with a maximum at 0 Hz, and some periodicity. The frequency response is zero at every multiple of 250Hz. Compare this with the frequency response of a unit pulse of 8ms in duration. Here the general shape of the signal is the same, but the zero crossings are at a spacing of 125Hz. These figures are the reciprocals of the pulse duration, indicating that there are inverse relationships between time and frequency. Generally, longer time periods relate to smaller frequency spans.

The formula for the frequency response of a unit pulse may be calculated directly from the Fourier transform equation as

$$X(f) = \int_{-T/2}^{T/2} e^{-j2\pi ft} dt = -\left[\frac{1}{j2\pi f} e^{-j2\pi ft} \right]_{-T/2}^{T/2} = \frac{\sin(\pi f T)}{\pi f} = T \operatorname{sinc}(\pi f T)$$

where T is the duration of the pulse. You can observe the changes in magnitude caused by the different values of T , as well as the changes in the spacing of the zero crossings, a function of the sin component.

Sinusoids and cosinusoids are signals that by definition contain only one frequency of signal. The tool has two examples of these with frequencies 333Hz and 500Hz. The time domain and frequency transform of a 500Hz cosine wave is given by the following equations

$$x(t) = \cos(2\pi 500t)$$

$$X(f) = \frac{1}{2} (\mathcal{A}f - 500) + \mathcal{A}f + 500)$$

Delaying a 500Hz cosine wave by 0.5ms results in a sine wave signal, and its transform can be seen to be

$$x(t) = \sin(2\pi 500t)$$

$$X(f) = \frac{1}{j2} (\mathcal{A}f - 500) - \mathcal{A}f + 500) = \frac{j}{2} (\mathcal{A}f + 500) - \mathcal{A}f - 500)$$

As this change is made, by adding the delay, you will observe that the phase of the frequency transform changes, but the magnitude remains the same. Alternatively, using the real and quadrature representation, components that were purely real before becoming imaginary after the delay.

Delay and phase change

Any of the signals can be advanced or delayed by a number of predefined delays of up to 4ms. Alternatively, you can delay a signal by an arbitrary amount by clicking and dragging the graph whilst holding down a key on the keyboard. In the frequency domain this relates to alteration of the phase of the signal, thus no difference will be observed when viewing the Fourier transform magnitude plot, but will be evident when viewing the phase of the transform or the real and imaginary parts together.

Try this out for various types of signal.

Take particular note of the scaled unit impulse as without any delay it results in a purely real transform of height 0.004 (the scaling factor). When this signal is delayed, the transform becomes a cosinusoid in the real component and a sinusoid in the imaginary. The formula for this is

$$X(f) = 0.004[\cos(2\pi f \tau) - j \sin(2\pi f \tau)] = 0.004e^{-j2\pi f \tau}$$

where τ is the value of the delay.

This implies that a delay of a specific amount in the time domain equates to multiplication by a phasor in the frequency domain. Set the delay for the scaled unit impulse to 0.5ms as was done for the 500Hz cosine waveform in the previous section. Now note the values of the real and imaginary parts of the transform at 500Hz and -500Hz. Now switch the input signal to the 500Hz cosine and you should be able to explain how the purely real transform of the undelayed waveform relates to the purely imaginary transform of the delayed signal.

Not only can the time domain signal be delayed, but the frequency transform can be shifted, resulting in a phase change in the time domain. Experiment with this observing the time domain signal as magnitude and phase, and as real and quadrature to see the effects that can be obtained. Try shifting the frequency response of a cosinusoid, or a sinusoid, so that one of the frequency samples is set to 0Hz. The result will be a complex phasor, consisting of a cosinusoid and sinusoid in the real and imaginary components of the time domain plus a DC offset from the 0Hz component.

Multiplication and convolution

Using the tool, review the transforms of the unit pulse function and the cosine function. For the moment it is best to view these using the magnitude and phase representation of the frequency domain.

Now switch to one of the 8ms segment of a cosine or sine waveforms. You should observe that the frequency domain plot is some form of combination of the two types of signal. Strictly speaking, the time domain signal is the multiplication of a unit pulse of 8ms duration delayed by 4ms, and a cosinusoid or sinusoid waveform of the selected frequency. The frequency domain transform is then the addition of two sinc functions which have been shifted in frequency. Notice where the highest peaks are and you should observe that these correspond with the frequency of the sine or cosine signal. What has happened is that in the frequency domain the sinc function from the unit pulse and the two impulses from the sine or cosine function have been convolved together. This is an example of the general rule that multiplication in the time domain equates to convolution in the frequency domain.

You can reconstruct the two constituent waveforms by shifting the frequency response of the 8ms unit pulse to 500Hz, and to -500Hz. You should find that the real component of the two shifted signals are the same, but that the quadrature components are the complement of each other. Thus when they are summed together, the result is a signal with a real component and a zero quadrature component.

In fact an equivalent rule also holds that convolution in the time domain equates to multiplication in the frequency domain. Thus, for example, a complex phasor in the frequency domain multiplied by a given signal's transform produces a time domain function where an impulse is convolved with signal. This is precisely what is happening when the delay value is being altered.

Summary

By working through this document you should understand the relationships between time and frequency domain. You should understand the following points

- Signal shape dictates the overall shape of the frequency transform

- Signal duration is related to the width of the frequency transform
- Sinusoidal signals contain single frequency components
- A time delay is equivalent to a linear phase change in frequency
- Multiplication in time is equivalent to convolution in frequency
- Convolution in time is equivalent to multiplication in frequency

Fourier Series to Transform

Using this tool you can select a variety of periodic signals. The signal is shown in the lower graph, and its Fourier series shown in the upper graph as a magnitude and phase representation.

The Fourier series is given by

$$X[k] = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-j\frac{2\pi kt}{T}} dt$$

where T is the period of the signal.

The animation buttons take the basic signal shape, and increase the period of the signal by spacing out the repetitions of the basic shape in time. For each click of the single step button, the period of the waveform is doubled. You can then observe that in the frequency representation, increasing the period introduces more spectral lines. Each doubling of the period introduces a spectral line between two currently existing lines. Repeatedly increasing the period results in a very dense frequency domain representation.

Note also that the scale of the frequency domain representation decreases as the period is increased. This is due to a reduction in power as the period is increased. (The basic signal shape has a fixed energy associated with it. As the period is increased, the energy in one period remains the same, but as the length has increased, the power decreases).

Taking this to its extreme, if the period is increased to infinity, making the signal aperiodic, then the spacing of the spectral lines becomes infinitely small, and a continuous frequency response results. The magnitude of the response will be zero, but from this derivation, the Fourier transform can be obtained by removing the normalization factor at the front of the Fourier series equation, and

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$$

Discrete Fourier Transform

Sampled in time and frequency

Starting with the 500Hz sine wave, you should observe that in the frequency domain there are two pulses, one at 500Hz, and one at the aliased frequency of 7500Hz. With the Fourier transform, a sine wave produced two impulses, i.e.

$$s(t) = \sin\left(\frac{2\pi t}{500}\right) \quad (1)$$

$$S(f) = \frac{1}{2j}(\delta(f - 500) - \delta(f + 500)) \quad (2)$$

Since the signal is sampled in the time domain, it includes periodic repetitions of (14) spaced by the sampling rate of 8kHz. Thus, pulses are seen at 500Hz and at 8000Hz-500Hz=7500Hz from the first of these periodic repetitions.

The reason that the DFT has only discrete pulses in the frequency domain, and not impulses of infinite height, is that it is assumed that the signal being transformed is periodic with a period equal to the *length* of the DFT ($T = N \Delta t$ where Δt is the sampling interval, and N is the number of samples). For this case $\Delta t = 1/8000 = 0.125\text{ms}$ and $N = 64$ when the DFT tool is first started, thus a period of 8ms is assumed.

Now try changing the input signal to a unit pulse of length 4ms. You can now clearly see that the frequency domain consists of a number of discrete components. If you use your mouse to click on the frequency graph to magnify a section and move it along the scale you should see that most of the components are spaced by 250Hz, except for the very first two with a spacing of 125Hz. In reality, all of the frequency components are spaced by 125Hz, it is just that every second component has a value of 0, hence the apparent 250Hz spacing.

This spacing, Δf is defined by the reciprocal of the period just calculated above, i.e.

$$\Delta f = \frac{1}{T} = \frac{1}{N \Delta t} = \frac{1}{64 \Delta t} = \frac{8000}{64} = 125\text{Hz} \quad (3)$$

You may wish to compare these transform with the 8ms segments of a 500Hz cosine wave and the pulse of 4ms duration in the Fourier transform tool to see how the waveform has been sampled in the DFT. If you examine the Fourier transform at multiples of 125Hz, you will find a correspondance between what is seen in the DFT frequency domain and the Fourier transform frequency domain. For the example of the 500Hz sinusoid/cosinusoid you can observe that all of the discrete frequencies that are multiples of 125Hz are zero, save for those at 500Hz and -500Hz. For the case of the unit pulse the non-zero values occur at 0Hz, and odd multiples of 125Hz only. Thus the DFT may be considered to be a sampled version of a Fourier transform that has been made periodic with a period of 8kHz in this example.

Frequency spacing and number of samples

The spacing of the samples in the frequency domain has been shown to be related to the sampling rate and number of samples in equation (3). If we wished to reduce the frequency spacing, Δf then there are two options for doing so. The first is to increase the sample Δt spacing by decreasing the sampling rate f_s . However this would move the periodic repetitions in frequency closer together. Alternatively the number of samples, N , can be increased to reduce the frequency domain sample spacing. However, if

only the 64 samples shown are available, how can additional samples be introduced. One method is to simply append 0 values to the end of the data set, hence increasing the total number of samples. This is termed zero padding. To see the effect of this click on the checkbox in the tool.

Each frame of the animation shows the result of doubling the total number of samples N by adding zeros to the end of the original data sequence. It is clear that every doubling of N results in a halving of Δf . It should now be clear for the two signals that we have already looked at that the DFT is strongly related to a sampled version of the Fourier transform as mentioned previously.

You may ask why bother with zero padding. One answer is that it allows you to ascertain frequencies with more accuracy. Switch the input signal to a 333Hz sine wave. With the zero padding on you can, by zooming into the frequency domain plot, see that the peak of the two highest ‘lobes’ lie at approximately -333Hz and 333Hz as expected. (Actually at this frequency spacing the highest pulses are at 336Hz and -336Hz). Now try this after switching zero padding off. You will find that the best estimate that can be made is that the frequency is around 375Hz. We will also find zero padding useful in explaining what happens with windowing.

It is also noticeable in the non-zero padded version that not only are there non-zero values at 375Hz and 7625Hz (being the closest values to 333Hz and 7667Hz that are represented in the DFT), all of the other samples are non-zero. Compare this with the Fourier transform of a 333Hz cosine wave and you should see that it has been sampled not at the zero amplitude values of the frequency domain as the 500Hz sine wave was, but at non-zero amplitudes for all the samples spaced by 125Hz. This effect in the DFT is called leakage as the power of the sine wave ‘leaks’ into other frequency bins. (A bin is simply a sample in the frequency domain, eg 375Hz is the fourth bin in the non-zero padded spectrum, and the 64th bin in the zero padded version).

You may wish to investigate at this stage what the zero padded versions of the other signals look like.

More complex signals

Now switch the source signal to the ‘333Hz sine wave plus low power signal’ option. You should see that there is very little difference between this signal and the 333Hz sine wave by itself. Even with zero padding little clear difference can be observed. It is clear that if there were any noise on the signal being measured that no distinction could be made between the two.

Now select a window such as a Hamming window from the selection of options. You will see the time domain signal being altered and the frequency domain plot will change at the same time. If zero padding is not being applied the change in the time domain can be clearly seen that both ends of the signal sample are attenuated, while samples closer to the center are attenuated by progressively smaller amounts. In actual fact, each sample is being multiplied by a factor determined by a raised cosine equation such that

$$w[i] = 0.54 + 0.46 \cos(2\pi i / (N - 1))$$

where $w[i]$ is the scaling factor for the i^{th} sample. The difference in the frequency plot is obvious. The low power signal of 1kHz can now be seen above the leakage of the 333Hz signal.

To explain what is happening, switch the source signal to an 8ms duration unit pulse, making sure that zero padding is switched on. Switch between no windowing and the Hamming window and you will see how the frequency domain changes. Remembering that multiplication in time is equivalent to convolution in frequency, it should become apparent that the frequency transform of the window shape is being convolved with the impulses of the two sine wave signals. The first of these is the 333Hz signal, and the second can be seen to be at 1kHz once a Hamming window is applied.

In reality, any sampled signal that is transformed is implicitly being windowed by a rectangular window since only a limited number of samples are used. This is why leakage is such a problem. However by application of other window shapes, the frequency domain representation of the sampled signal can be modified to highlight certain features. Try other window shapes on the 8ms unit pulse and the mixture of sine waves to see what each window does to the frequency domain transform.

Summary

After using the DFT tool you should understand the following points:

- The DFT is sampled in both time and frequency domains
- The DFT is therefore periodic in both time and frequency domains
- Zero padding in the time domain can be used to reduce sample spacing in the frequency domain
- Application of the DFT to real world signals results in leakage in the frequency domain as only a limited number of samples is used
- The effects of leakage in the frequency domain can be altered by multiplying the time domain signal by a window shape