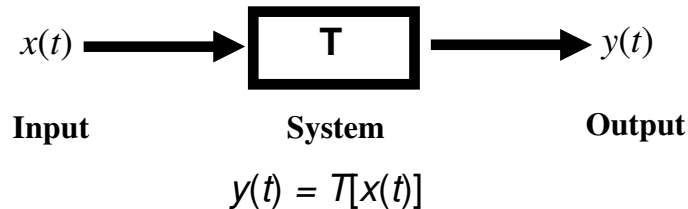


Linear Time-invariant systems, Convolution, and Cross-correlation

(1) Linear Time-invariant (LTI) system

A **system** takes in an input function and returns an output function.



An **LTI system** is a special type of system. As the name suggests, it must be both *linear* and *time-invariant*, as defined below.

LINEAR

Scaling: $T[ax(t)] = aT[x(t)]$

Superposition: $T[x_1(t) + x_2(t)] = T[x_1(t)] + T[x_2(t)]$

TIME INVARIANT

If $y(t) = T[x(t)]$, then $y(t-s) = T[x(t-s)]$

Many systems in neuroscience can be approximated as LTI:

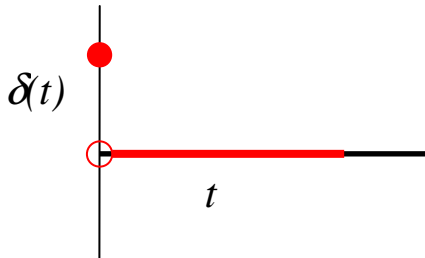
Input	LTI system	Output
Pre-synaptic action potentials	Synapse	Post-synaptic conductance
Visual stimulus	Eye	Retinal image
Stimulus contrast	Retinal ganglion cell	Firing rate
Injected Current	Passive neural membrane	Membrane potential

The reason LTI systems are incredibly useful is because of a key fact: if you know the response of the system to an impulse, then you can calculate the response of the system to ANY input. This gives you an enormous amount of predictive power.

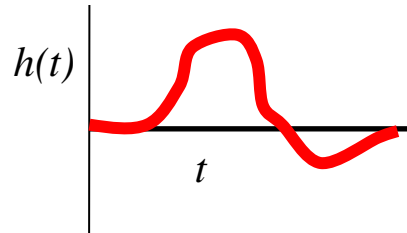
But what do I mean by an impulse? As plotted in the figure below, the impulse function is just a function that is zero at all times except for at $t=0$ (left panel below). You can think of it as the force on an object created by a hammer hitting it – the force is 0 except for at the moment of impact. The impulse response is the response of the system to the impulse. You can think of this as the response of your leg to your knee getting hit with a hammer at the doctor's office. The right panel below is an example of what the impulse response of a system might look like.

The Impulse *function*:

$$\delta(t) = \begin{cases} 1, & t = 0 \\ 0, & t \neq 0 \end{cases}$$



The Impulse *response*:



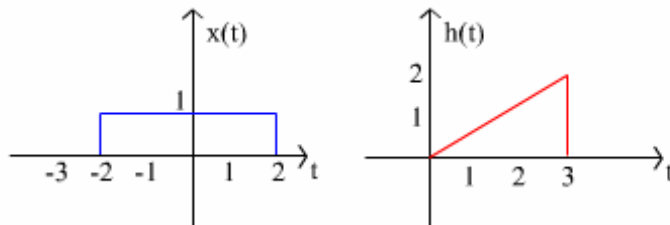
(2) Convolution

How do we predict the response of our system to an arbitrary input once we know the impulse response? The secret is **convolution**. It allows us to predict the response of our knee to some arbitrary input function (once we know the response to the hammer), or the response of a building to an earthquake (once we know how it responds to the impact of an impulse).

Convolution is a mathematical operation which takes two functions and produces a third function that represents the amount of overlap between one of the functions and a reversed and translated version of the other function. Here is the mathematical definition of convolving two functions, $x(t)$ and $h(t)$, to create an output $y(t)$:

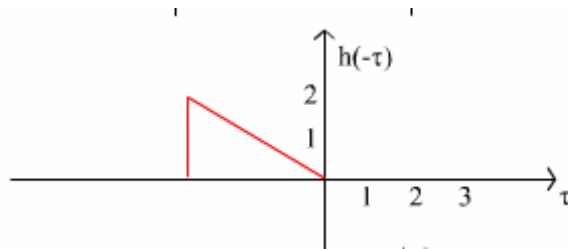
$$y(t) = \sum_{\tau=-\infty}^{\infty} x(\tau)h(t-\tau)$$

This is often written in shorthand as $y=x*h$, where the $*$ represents the convolution operation. You can think of t in the equation above as a variable that represents time. But what is τ ? It's called a "dummy" variable. As you can see, it doesn't exist on the left hand side of the equation – it's just used to index over time for the integration on the right hand side. Hopefully, the example below of convolving sample functions $x(t)$ and $h(t)$ will make this clear¹. Here are our functions $x(t)$ and $h(t)$.

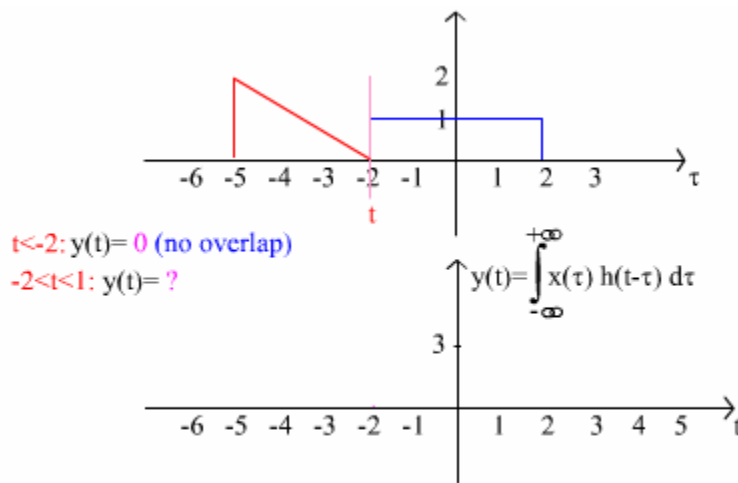


¹ Example borrowed from http://www.ee.washington.edu/class/235dl/EE235/Project/lesson6/lesson6_1.html

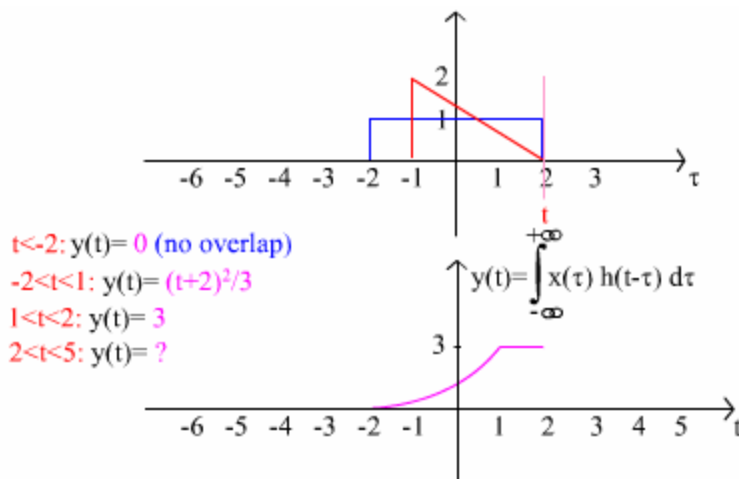
The first step is to reverse one of the functions. We choose to reverse $h(t)$, which we now plot on the “dummy” time axis τ .



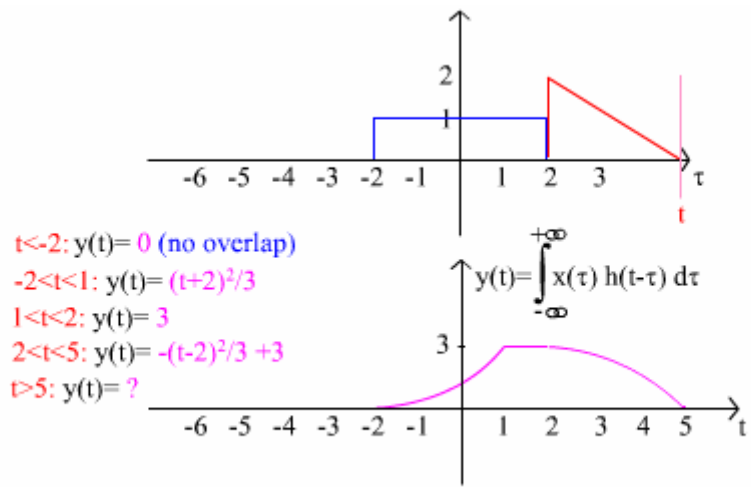
We plot $x(\tau)$ on the same axis, and begin the process of shifting $h(-\tau)$ by t , and comparing it to $x(\tau)$. Since these are continuous (not discrete) functions, we take an integral (not the sum) when calculating the convolution. In the figure below, h is shifted by $t=-2$. For this value of shift, there is no overlap between $x(\tau)$ and $h(t-\tau)$, so $y(t)=0$ (which is why it doesn't show up in the plot below).



As we continue to shift $h(t-\tau)$ by changing t , there gets to be overlap between the two functions, as plotted below (the convolution, $y(t)$, is depicted in pink). The overlap is quantified by multiplying the two input functions on a point by point basis, and then integrating the resulting function.



Once t reaches 5, there is no longer overlap between the two functions, and $y(t)=0$ once again.



Convolution is an incredibly useful operation because it can be used to predict the output of an LTI system to *any* input. Just think of $x(t)$ as the arbitrary input function (e.g. a visual stimulus) and $h(t)$ as the response of the system to an impulse (e.g. the firing rate of the retinal ganglion cell to a flash of light). Then the convolution of $x(t)$ and $h(t)$ is the predicted output of the system (e.g. the firing rate in response to the arbitrary visual stimulus). Because of this great predictive power, LTI systems are used all the time in neuroscience.

A useful thing to know about convolution is the **Convolution Theorem**, which states that convolving two functions in the time domain is the same as multiplying them in the frequency domain:

If $y(t) = x(t) * h(t)$, (remember, $*$ means convolution)
then $Y(f) = X(f)H(f)$ (where Y is the fourier transform of y , X is the fourier transform of x , etc)

Since multiplying is a simpler operation than convolving, it is often easier to work in the frequency domain.

(3) Cross-correlation

Cross-correlation is a mathematical operation that is used to quantify the similarity of two functions for different time delays. For example, it can be used to compare firing rate between two neurons to see at what relative time delay the firing is most similar.

Despite having a different purpose than convolution, the mathematical operation of cross-correlation is very similar to that of convolution. Mathematically, the main difference is that when two functions are cross-correlated, neither function is time-reversed before being shifted and compared.

Cross-correlation

$$f \star g = \int_{-\infty}^{\infty} \overline{f(\tau)} g(t + \tau) d\tau$$

Convolution

$$f * g = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau$$

As a simple example, let's consider the cross-correlation of two neurons with the following spiking pattern (1 represents a spike during the timebin; 0 represents no spike).

time
 $\xrightarrow{\hspace{10em}}$

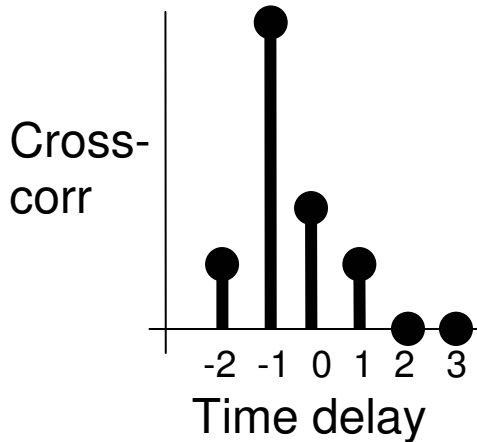
Neuron 1 spike train: 0 0 1 0 1 0 0 0 1 1 0 0 0 1 1

Neuron 2 spike train: 0 0 0 1 0 1 0 0 0 1 1 0 0 0 1

To calculate the cross-correlation, we need to compare the spike trains for different relative time delay (i.e. different relative shifts of the two spike trains). First, let's consider a time delay of 0. When we multiply the two spike trains on a point by point basis, we get the following:

0 0 0 0 0 0 0 0 0 1 0 0 0 0 1

The next step of calculating the cross correlation is to sum across all these values. This gives us a cross-correlation of 2 (for a time delay of 0). We repeat the same calculation for different delays. As you can see from the plot below, the cross correlation peaks for time delay of -1.



The interpretation of this cross-correlation is that the two spike trains are in the greatest agreement when spike train 2 is shifted backwards in time by one time point. In other words, neuron 2 tends to spike after neuron 1. A biological scenario that could lead to this cross-correlation profile is that neuron 2 is postsynaptic to neuron 1.