• The convolution of two signals in the time domain is equivalent to multiplication in the frequency domain, and vice versa, i.e. the spectrum of two convolved signals is simply the product of the spectra of those signals, and the spectrum of two signals that have been multiplied together is the convolution of the spectrum of those two signals (Section 3.3.6).

• Let us define the bandwidth of a signal as the size of the set of range of frequencies the spectrum of the signal occupies. Hence, a signal that has frequency components from 1 MHz to 1.5 MHz, and no components outside this range, has a bandwidth of 1.5-1 = 0.5 MHz, or 500 KHz. Often, signals have a very large actual bandwidth, or even infinite bandwidth (such as in the case of a rectangular pulse), but have a lot of power concentrated in just a few frequencies. In this case, we can define the bandwidth more approximately, such as, the range of frequencies over which 90% of the signal is concentrated, so that if we used a filter to remove all frequencies outside this range, we would still get a good approximation for the signal.

• Usually, when we convolve two functions, the resulting function will have a width (or duration) equal to the sum of the widths of the original functions. When we multiply two signals together, we are convolving their Fourier spectra, so that the bandwidth of the product of signals is roughly equal to the sums of the bandwidths of each signal. So, $g^2(t)$ occupies twice the bandwidth of $g(t)$, $g^3(t)$ occupies three times the bandwidth, and so on.

• Let $x(t)$ be the input to a filter, and $y(t)$ the output. Then, for a linear time invariant filter, $y(t) = x(t) * h(t)$, where $h(t)$ is the “impulse response” of the signal, i.e., it is the output of the filter when the input is a delta function $\delta(t)$. To check this, let us set $x(t) = \delta(t)$:

$$
y(t) = \delta(t) * h(t) = \int_{-\infty}^{t} \delta(t-\tau)h(\tau)d\tau \quad \text{(from the definition of convolution)}$$

$$= \delta(t-\tau)h(\tau)|_{\tau=t} \quad \text{(because } \delta(t-\tau) \text{ has a non-zero value only when } \tau=t)$$

$$= h(t)$$

We have also seen here that convolution with the delta function yields the original function itself. We can also speak of the Fourier Transform of $h(t)$, namely, $H(f)$, the frequency response of the filter. Then, $Y(f) = X(f)H(f)$.

• Since the Fourier Transform is complex in general, we can also write $H(f) = |H(f)|e^{j\Theta(f)}$, where $|H(f)|$ is called the magnitude response of the filter, and $\Theta(f) = \angle H(f)$ is the phase response of a filter. The spectrum of the output of a filter is obtained only by first multiplying the input spectrum by $|H(f)|$, and then by $e^{j\Theta(f)}$.

• If $\Theta(f) = -2\pi t_0 f$, i.e., the phase of the output has a linear relationship with the input frequency, we call such a filter a linear phase filter. This means that lower frequencies undergo a small phase shift, and higher frequencies undergo a larger phase shift. The net effect of this is that all frequencies are delayed by an equal amount in the time domain (see Figure 3.19 in the text for why this should be so). We can also understand this from the time-shifting property: recall that multiplying by $e^{-j2\pi t_0 f}$ is equivalent to shifting a signal by $t_0$ in the time domain. That is all that is going on here.
• From a plot of $\Theta(f) = -2\pi t_0 f$, we can tell whether there will be constant time delay for all frequencies (which is how it will be, if the plot is a straight line), and conclude how long the delay is from the slope of the plot. Higher the slope, larger the value of $t_0$, so longer the delay. These details are discussed in Section 3.4.

• A filter that only lets frequencies less than some frequency $f_c$ Hz through is called a low pass filter (LPF). Usually, when we say it allows frequencies less than $f_c$ through, we mean that it lets these frequencies through on the positive frequency side, and has a mirror image response on the negative side. So, an ideal LPF will have the response:

$$H(f) = \begin{cases} 1 & \text{if } |f| < f_c \\ 0 & \text{otherwise} \end{cases}.$$  

Here, we have $\Theta(f) = 0$ for all $f$ (no delay in the output). If we had $\Theta(f) = -j2\pi ft_0$, this

Figure 1:

would mean that the output signal will be delayed by an additional $t_0$ seconds. Using the notation from previous classes, you will notice that for the LPF, $H(f) = \Pi(f/(2f_c))$. Similar to the LPF, we have high pass filters (HPF), band pass filters (BPF), and band stop filters (BSF).

• Section 3.3.7 explains how differentiation in the time domain is equivalent to multiplying by $2\pi jf$ in the frequency domain. This is equivalent to multiplying by a filter of frequency response equal to $2\pi jf$, which has a magnitude response growing linearly with $f$. If you plot this magnitude response vs $f$, it is a triangle shaped function, which filters out lower frequencies and passes higher frequencies. This is the shape of a high pass filter. So differentiation is nothing but a kind of high pass filtering. Similarly, integration is a kind of low pass filtering. The frequency domain equivalent of integration is also given in Section 3.3.7.

• For an ideal LPF, the impulse response $h(t) = F^{-1}[H(f)] = F^{-1}[\Pi(f/(2f_c))] = 2f_c \text{sinc}(2\pi tf_c)$, from the Class 3 guide. If we were to implement this in real life, this poses two problems: the impulse response has an infinite duration, and the impulse response acts on negative time, i.e., $h(t) \neq 0$ for $t < 0$. A filter that acts on negative time essentially needs to act on an input that has not even been applied yet. Clearly, this is impractical, and such filters are called non-causal filters. Hence, only causal filters, where $h(t) = 0$ for $t < 0$ are realizable.

• To make a non-causal filter realizable, we can chop off the portion for $t < 0$ and make it zero. For a low pass filter, this portion has significant power (half the power of the sinc function is in the negative-$t$ axis), but we can delay the impulse response by, say, time $t_d$, which will
shift the sinc function to the right, and then chop of the portion for \( t < 0 \). This means that the LPF can be made realizable, but suffers the downside of delaying the output by time \( t_d \).

Theoretically, if we were willing to wait an infinite amount of time for our output, we could build a perfect filter. Usually, however, \( t_d \approx 4/(2B) \) to \( 3/(2B) \), where \( B \) is the bandwidth of the input signal, is sufficient. For example, audio signals have \( B = 20 \) KHz, which corresponds to about a 1 mS delay, which is acceptable. All this is discussed in Section 3.5.

- Recall that \( \mathcal{F}[\cos(2\pi f_0t)] = \frac{1}{2} (\delta(f - f_0) + \delta(f + f_0)) \), from the Class 3 guide. By the symmetry property, since the cosine signal is real, you will notice that the magnitude spectrum in the negative frequency region is simply the mirror image of the positive frequency region. As long as we add, multiply and square cosine signals, and pass it through filters that do not alter its phase (or even if they alter the phase, but are linear phase), we will still only be dealing with real signals, so the mirror image property of the Fourier spectra will hold true.

- Recall that multiplying by \( e^{j2\pi f_0t} \) in the time domain, is equivalent to shifting the Fourier spectrum by \( f_0 \). Since \( \cos(2\pi f_0t) = 1/2(e^{j2\pi f_0t} + e^{-j2\pi f_0t}) \), the product of any signal with a cosine signal will yield a spectrum that is the sum of two frequency-shifted spectra. That is, for any signal \( x(t) \):

\[
\mathcal{F}[x(t) \cos(2\pi f_0t)] = \mathcal{F}\left[x(t) \frac{e^{j2\pi f_0t} + e^{-j2\pi f_0t}}{2}\right]
= \mathcal{F}\left[x(t) e^{j2\pi f_0t}\right] + \mathcal{F}\left[x(t) e^{-j2\pi f_0t}\right]
= \frac{1}{2} (X(f - f_0) + X(f + f_0))
\]

We can also derive the same result using the convolution property:

\[
\mathcal{F}[x(t) \cos(2\pi f_0t)] = \mathcal{F}[x(t)] * \mathcal{F}[\cos(2\pi f_0t)]
= X(f) * \frac{1}{2} (\delta(f - f_0) + \delta(f + f_0))
= \frac{1}{2} (X(f) * \delta(f - f_0) + X(f) * \delta(f + f_0))
= \frac{1}{2} (X(f - f_0) + X(f + f_0)),
\]

where we get the last step from the fact that convolution with a delta function yields the function itself.

- Of particular interest, is the case when \( x(t) = \cos(2\pi f_1t) \) in the above example, i.e., a pure cosine signal. Since \( X(f) = (1/2)(\delta(f - f_0) + \delta(f + f_0)) \) in this case, we can substitute this in the above result to get:

\[
\mathcal{F}[\cos(2\pi f_1t) \cos(2\pi f_0t)] = \frac{1}{4} (\delta(f - f_1 - f_0) + \delta(f - f_1 + f_0) + \delta(f + f_1 - f_0) + \delta(f + f_1 + f_0))
\]

Now, we can easily take the inverse Fourier transforms of delta functions (from the Class 3 guide):

\[
\mathcal{F}^{-1}\left[\frac{1}{4} (\delta(f - f_1 - f_0) + \delta(f - f_1 + f_0) + \delta(f + f_1 - f_0) + \delta(f + f_1 + f_0))\right]
\]
In other words, when we multiply cosine signals, we get a sum of two cosine signals of frequencies $f_1 + f_0$ and $f_1 - f_0$, respectively, each of amplitude equal to half of the original amplitude. When we plot the deltas corresponding to the spectrum of a cosine signal of amplitude $A$, it will give rise to two delta functions (one on the positive side, one on the negative frequency side), each with magnitude $A/2$. The cosine is the sum of these two deltas, so its amplitude will be $A$.

**What happens if we add $\cos(2\pi f_2 t)$ to the product $\cos(2\pi f_1 t) \cos(2\pi f_0 t)$, i.e., what is the spectrum of $\cos(2\pi f_2 t) + \cos(2\pi f_1 t) \cos(2\pi f_0 t)$?** Since the product $\cos(2\pi f_1 t) \cos(2\pi f_0 t)$ has frequencies $f_1 + f_0$ and $f_1 - f_0$ and their mirror images on the negative frequency axis, each of amplitude $1/4$, by adding $\cos(2\pi f_2 t)$, which has only one frequency $f_2$ (and its mirror image) of amplitude $1/2$, we simply add the Fourier spectra together (from linearity of the Fourier transform). So, the result will have frequencies $f_1 + f_0$, $f_1 - f_0$, and $f_2$ (as well as their mirror images on the negative frequency axis). That is,

$$
\mathcal{F}[\cos(2\pi f_2 t) + \cos(2\pi f_1 t) \cos(2\pi f_0 t)] = \mathcal{F}[\cos(2\pi f_2 t)] + \mathcal{F}[\cos(2\pi f_1 t) \cos(2\pi f_0 t)]
$$
\[
\frac{1}{2} \left( \delta(f - f_2) + \delta(f + f_2) \right) + \frac{1}{4} \left( \delta(f - f_1 - f_0) + \delta(f - f_1 + f_0) \right) + \frac{1}{4} \left( \delta(f + f_1 - f_0) + \delta(f + f_1 + f_0) \right)
\]

- What happens if we multiply \( \cos(2\pi f_2 t) \) by \( \cos(2\pi f_1 t) \cos(2\pi f_0 t) \)? We will get delta functions at four frequencies: \( f_1 + f_0 + f_2 \), \( f_1 + f_0 - f_2 \), \( f_1 - f_0 + f_2 \), and \( f_1 - f_0 - f_2 \), as well as their mirror images on the negative frequency axis, where each delta function is of magnitude \( 1/4 \times 1/2 = 1/8 \).

- What happens if we add a low pass filter to \( \cos(2\pi f_1 t) \cos(2\pi f_0 t) \)? Let us assume that the cutoff frequency for the LPF \( f_c \) is between \( f_1 - f_0 \) and \( f_1 + f_0 \). In this case, the LPF will only pass frequencies where \( |f| < f_c \), so it will pass \( f_1 - f_0 \) and its mirror image in the negative frequency axis through, but remove \( f_1 + f_0 \).

- A neat application of the above two results is to recover a signal that has been multiplied by a cosine signal. For example, if we multiply \( \cos(2\pi f_1 t) \cos(2\pi f_0 t) \) by \( \cos(2\pi f_2 t) \), where \( f_2 = f_0 \), we will get the frequencies \( f_1 + 2f_0 \), \( f_1 - 2f_0 \) and \( f_1 \) (which has twice the magnitude of the other 2 components, if you are careful to notice). If we apply a low pass filter with \( f_c \) in-between \( f_1 \) and \( f_1 + 2f_0 \), we will reject all terms except the \( f_1 \) term. Thus, we have recovered the one of the original frequency components we started out with, namely \( f_1 \).

- With the above results and examples, we should be generally comfortable with what happens when we add, multiply and filter (with ideal filters) cosine signals, and be able identify the frequency components of the result without actually having to do all the math. We should also be able to sketch the resulting spectra, and even if we have trouble getting the exact magnitudes of the resulting frequencies, we can at least tell which frequencies are present in the output.

- We also dealt with four kinds of distortion: time dispersion (Section 3.6.1), frequency dispersion (Section 3.6.2), delays (Section 3.6.3), and fading (Section 3.6.3).