MIT 6.02 **DRAFT** *Lecture Notes*
Fall 2011 (Last update: November 9, 2011)
Comments, questions or bug reports?
Please contact verghese at mit.edu

CHAPTER 12
Convolution and Frequency Response for LTI Systems

■ 12.1 A Second Look at Convolution

As we have seen in the previous chapter, a discrete-time (DT) LTI system that maps an input signal $x[.]$ to an output signal $y[.]$ is completely characterized by its response to a *unit sample function* (or unit pulse function, or unit "impulse" function) $\delta[n]$ at the input. Recall that $\delta[n]$ takes the value 1 where its argument $n = 0$, and the value 0 for all other values of the argument. An alternative notation for this signal that is sometimes useful for clarity is $\delta_0[.]$, where the subscript indicates the time instant for which the function takes the value 1; thus $\delta[n-k]$, when described as a function of n, could also be written as the signal $\delta_k[.]$.

The **unit sample response** $h[n]$ is simply the sequence of values that $y[n]$ takes when we set $x[n] = \delta[n]$, i.e., $x[0] = 1$ and $x[k] = 0$ for $k \neq 0$. The response $h[n]$ to the elementary input $\delta[n]$ can be used to characterize the response of an LTI system to *any* input, for the following two reasons:

- An arbitrary signal $x[.]$ can be written as a sum of scaled (or weighted) and shifted unit sample functions. This is expressed in two ways below:

$$\begin{aligned} x[.] &= \cdots + x[-1]\delta_{-1}[.] + x[0]\delta_0[.] + \cdots + x[k]\delta_k[.] + \cdots \\ x[n] &= \cdots + x[-1]\delta[n+1] + x[0]\delta[n] + \cdots + x[k]\delta[n-k] + \cdots \end{aligned} \tag{12.1}$$

- The response of an LTI system to an input that is the scaled and shifted combination of other inputs is the same scaled combination—or **superposition**—of the correspondingly shifted *responses* to these other inputs.

Since the response at time n to the input signal $\delta[n]$ is $h[n]$, it follows from the two observations above that the response at time n to the input $x[.]$ is, see Slide 12.2,

$$\begin{aligned} y[n] &= \cdots + x[-1]h[n+1] + x[0]h[n] + \cdots + x[k]h[n-k] + \cdots \\ &= \sum_{k=-\infty}^{\infty} x[k]h[n-k] \ . \end{aligned} \tag{12.2}$$

Example 1 Suppose $h[n] = (0.5)^n u[n]$, where $u[n]$ denotes the unit step function defined previously (taking the value 1 where its argument n is non-negative, and the value 0 when the argument is strictly negative). If $x[n] = 3\delta[n] - \delta[n-1]$, then

$$y[n] = 3(0.5)^n u[n] - (0.5)^{n-1} u[n-1] \ .$$

From this we deduce, for instance, that $y[n] = 0$ for $n < 0$, and $y[0] = 3$, $y[1] = 0.5$, $y[2] = (0.5)^2$, and in fact $y[n] = (0.5)^n$ for all $n > 0$.

The above example illustrates that if $h[n] = 0$ for $n < 0$, then the system output cannot take nonzero values before the input takes nonzero values. Conversely, if the output never takes nonzero values before the input does, then it must be the case that $h[n] = 0$ for $n < 0$. In other words, this condition is necessary and sufficient for **causality** of the system.

Example 2 (Scale-&-Delay System) Consider the system $\mathcal{S}$ in Slide 12.3 that scales its DT input by A and delays it by $D > 0$ units of time (or, if D is negative, advances it by $-D$). This system is linear and time-invariant (as is seen quite directly by applying the definitions from Chapter 10). It is therefore characterized by its unit sample response, which is

$$h[n] = A\delta[n-D] \ .$$

We already know from the definition of the system that if the input at time n is $x[n]$, the output is $y[n] = Ax[n-D]$, but let us check that the general expression in Equation (12.2) gives us the same answer:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} x[k]A\delta[n-k-D] \ .$$

As the summation runs over k, we look for the unique value of k where the argument of the unit sample function goes to zero, because this is the only value of k for which the unit sample function is nonzero (and in fact equal to 1). Thus $k = n - D$, so $y[n] = Ax[n-D]$, as expected.

A general unit sample response $h[.]$ can be represented as a sum—or equivalently, a parallel combination—of scale-&-delay systems, see Slides 12.3, 12.4:

$$h[n] = \cdots + h[-1]\delta[n+1] + h[0]\delta[n] + \cdots + h[k]\delta[n-k] + \cdots \ . \tag{12.3}$$

An input signal $x[n]$ to this system gets scaled and delayed by each of these terms, with the results added to form the output. This way of looking at the LTI system response yields the expression

$$\begin{aligned} y[n] &= \cdots + h[-1]x[n+1] + h[0]x[n] + \cdots + h[m]x[n-m] + \cdots \\ &= \sum_{m=-\infty}^{\infty} h[m]x[n-m] \ . \end{aligned} \tag{12.4}$$

A simple change of variables, setting $n - k = m$, shows that the expressions in Equations (12.2) and (12.4) are indeed equivalent, see Slide 12.5.

■ 12.1.1 Flip-Slide-Dotting Away: Implementing Convolution

The above descriptions of convolution explain why we end up with the expressions in Equations (12.2) and (12.4) to describe the output of an LTI system in terms of its input and unit sample response. We will now describe a graphical construction, Slide 12.6, that helps to visualize and implement these computations, and that is often the simplest way to think about the effects of convolution.

Let's examine the expression in Equation (12.2), but the same kind of reasoning works for Equation (12.4). Our task is to implement the computation in the summation below:

$$y[n_0] = \sum_{k=-\infty}^{\infty} x[k]h[n_0 - k] \ . \tag{12.5}$$

We've written n_0 rather than the n we used before just to emphasize that this computation involves summing over the dummy index k, with the other number being just a parameter, fixed throughout the computation.

We first plot the time functions $x[k]$ and $h[k]$ on the k axis (with k increasing to the right, as usual!)[1]. How do we get $h[n_0 - k]$ from this? First note that $h[-k]$ is obtained by reversing $h[k]$ in time, i.e., a **flip** of the function across the time origin. To get $h[n_0 - k]$, we now **slide** this reversed time function, $h[-k]$, to the *right* by n_0 steps if $n_0 \geq 0$, or to the left by $-n_0$ steps if $n_0 < 0$. To confirm that this prescription is correct, note that $h[n_0 - k]$ should take the value $h[0]$ at $k = n_0$.

With these two steps done, all that remains is to compute the sum in Equation (12.5). This sum takes the same form as the familiar **dot product** of two vectors, one of which has $x[k]$ as its kth component, and the other of which has $h[n_0 - k]$ as its kth component. The only twist here is that the vectors could be infinitely long. So what this steps boils down to is taking an instant-by-instant product of the time function $x[k]$ and the time function $h[n_0 - k]$ that your preparatory "flip and slide" step has produced, then summing all the products.

At the end of all this (and it perhaps sounds more elaborate than it is, till you get a little practice), what you have computed is the value of the convolution for the *single* value n_0. To compute the convolution for another value of the argument, say n_1, you repeat the process, but sliding by n_1 instead of n_0.

[1] Does the time axis go from right to left when this material is taught in languages that write from right to left?

To implement the computation in Equation (12.4), you do the same thing, except that now it's $h[m]$ stays as it is, while $x[m]$ gets flipped and slid by n to produce $x[n-m]$, after which you take the dot product. Either way, the result is evidently the same.

Example 1 revisited Suppose again that $h[m] = (0.5)^m u[m]$ and $x[m] = 3\delta[m] - \delta[m-1]$. Then

$$x[-m] = -\delta[-m-1] + 3\delta[-m] \ ,$$

which is nonzero only at $m = -1$ and $m = 0$. (Sketch this!) As a consequence, sliding $x[-m]$ to the *left*, to get $x[n-m]$ when $n < 0$, will mean that the nonzero values of $x[n-m]$ have *no overlap* with the nonzero values of $h[m]$, so the dot product will yield 0. This establishes that $y[n] = (x * h)[n] = 0$ for $n < 0$, in this example.

For $n = 0$, the only overlap of nonzero values in $h[m]$ and $x[n-m]$ is at $m = 0$, and we get the dot product to be $(0.5)^0 \times 3 = 3$, so $y[0] = 3$.

For $n > 0$, the only overlap of nonzero values in $h[m]$ and $x[n-m]$ is at $m = n-1$ and $m = n$, and the dot product evaluates to

$$y[n] = -(0.5)^{n-1} + 3(0.5)^n = (0.5)^{n-1}(-1 + 1.5) = (0.5)^n \ .$$

So we have completely recovered the answer we obtained in Example 1. For this example, our earlier approach—which involved directly thinking about superposition of scaled and shifted unit sample responses—was at least as easy as the graphical approach here, but in other situations the graphical construction can yield more rapid or direct insights.

■ 12.1.2 Deconvolution

We've seen in the previous chapter, specifically in Slides 11.12–11.24, how having an LTI model for a channel allows us to predict or analyze the distorted output $y[n]$ of the channel, in response to a superposition of alternating positive and negative steps at the input $x[n]$, corresponding to a rectangular-wave baseband signal. That analysis was carried out in terms of the unit step response, $s[n]$, of the channel.

We now briefly explore one plausible approach to *undoing* the distortion of the channel, assuming we have a good LTI model of the channel. This discussion is most naturally phrased in terms of the unit sample response of the channel rather than the unit step response. The idea is to process the received baseband signal $y[n]$ through an LTI system, or LTI *filter*, that is designed to cancel the effect of the channel.

Consider, as in the example of Slide 12.7, a channel that we model as LTI with unit sample function

$$h_1[n] = \delta[n] + 0.8\delta[n-1] \ .$$

This is evidently a causal model, and we might think of the channel as one that transmits perfectly and instantaneously along some direct path, and also with a one-step delay and some attenuation along some echo path.

Suppose our receiver filter is to be designed as a causal LTI system with unit sample response

$$h_2[n] = h_2[0]\delta[n] + h_2[1]\delta[n-1] + \cdots + h_2[k]\delta[n-k] + \cdots \ . \qquad (12.6)$$

Its input is $y[n]$, and let us label its output as $z[n]$. What conditions must $h_2[n]$ satisfy if we are to ensure that $z[n] = x[n]$ for all inputs $x[n]$, i.e., if we are to undo the channel distortion?

An obvious place to start is with the case where $x[n] = \delta[n]$. If $x[n]$ is the unit sample function, then $y[n]$ is the unit sample response of the channel, namely $h_1[n]$, and $z[n]$ will then be given by $z[n] = (h_2 * h_1)[n]$. In order to have this be the input that went in, namely $x[n] = \delta[n]$, we need

$$(h_2 * h_1)[n] = \delta[n] \ . \tag{12.7}$$

And if we satisfy this condition, then we will actually have $z[n] = x[n]$ for arbitrary $x[n]$, because

$$z = h_2 * (h_1 * x) = (h_2 * h_1) * x = \delta_0 * x = x \ ,$$

where $\delta_0[.]$ is our alternative notation for the unit sample function $\delta[n]$. The last equality above is a consequence of the fact that convolving any signal with the unit sample function yields that signal back again; this is in fact what Equation (12.1) expresses.

The condition in Equation (12.7) ensures that the convolution carried out by the channel is inverted or undone, in some sense, by the filter. We might say that the filter **deconvolves** the output of the system to get the input (but keep in mind that it does this by a further convolution!). In view of Equation (12.7), the function $h_2[.]$ is also termed the *convolutional inverse* of $h_1[.]$, and vice versa.

So how do we find $h_2[n]$ to satisfy Equation (12.7)? It's **not** by a simple division of any kind (though when we get to doing our analysis in the frequency domain shortly, it will indeed be as simple as division). However, applying the "flip–slide–dot product" mantra for computing a convolution, we find the following equations for the unknown coefficients $h_2[k]$:

$$\begin{aligned}
1 \cdot h_2[0] &= 1 \\
0.8 \cdot h_2[0] + 1 \cdot h_2[1] &= 0 \\
0.8 \cdot h_2[1] + 1 \cdot h_2[2] &= 0 \\
&\cdots \\
0.8 \cdot h_2[k-1] + 1 \cdot h_2[k] &= 0 \\
&\cdots \ ,
\end{aligned}$$

from which we get $h_2[0] = 1$, $h_2[1] = -0.8$, $h_2[2] = -0.8h_2[1] = (-0.8)^2$, and in general $h_2[k] = (-0.8)^k u[k]$.

Deconvolution as above would work fine if our channel model was accurate, and if there was no noise in the channel. Even assuming the model is sufficiently accurate, note that any noise process $w[.]$ that adds in at the output of the channel will end up adding $v[n] = (h_2 * w)[n]$ to the noise-free output, which is $z[n] = x[n]$. This added noise can completely overwhelm the solution. For instance, if both $x[n]$ and $w[n]$ are unit samples, then the output of the receiver's deconvolution filter has a noise-free component of $\delta[n]$ and an additive noise component of $(-0.8)^k u[k]$ that dwarfs the noise-free part. After we've understood how to think about LTI systems in the frequency domain, it will become much clearer why such deconvolution is so sensitive to noise.

12.2 Sinusoidal Inputs and Frequency Response

Sinusoids—and their close relatives, the complex exponentials—play a distinguished role in the study of LTI systems. The reason is that a sinusoidal input gives rise to a *sinusoidal output* again, and at the *same frequency* as that of the input. This is not a property that is obvious from anything we have said so far about LTI systems. Only the amplitude and phase of the sinusoid might be, and generally are, modified from input to output, in a way that is captured by the *frequency response* of the system, which we introduce in this section.

Before focusing on sinusoidal inputs, consider an input that is *periodic* but not necessarily sinusoidal, Slide 12.10. A signal $x[n]$ is periodic if

$$x[n+P] = x[n] \quad \text{for all } n \; ,$$

where P is some fixed positive integer. The smallest positive integer P for which this condition holds is referred to as the *period* of the signal (though the term is also used at times for positive integer multiples of P), and the signal is called P-periodic.

While it may not be obvious that sinusoidal inputs to LTI systems give rise to sinusoidal outputs, it's not hard to see that *periodic* inputs to LTI systems give rise to periodic outputs of the same period (or an integral fraction of the input period). The reason is that if the P-periodic input $x[.]$ produces the output $y[.]$, then time-invariance of the system means that shifting the input by P will shift the output by P. But shifting the input by P leaves the input unchanged, because it is P-periodic, and therefore must leave the output unchanged, which means the output must be P-periodic. (This argument actually leaves open the possibility that the period of the output is P/K for some integer K, rather than actually P-periodic, but in any case we will have $y[n+P] = y[n]$ for all n.)

12.2.1 Sinusoidal Inputs

A DT sinusoidal input takes the form

$$x[n] = \cos(\Omega_0 n + \theta_0) \; , \tag{12.8}$$

as in Slide 12.11. We refer to Ω_0 as the *angular frequency* of the sinusoid, measured in radians/sample; Ω_0 is the number of radians by which the argument of the cosine increases when n increases by 1.

Note that the *lowest* rate of variation possible for a DT signal is when it is constant, and this corresponds, in the case of a sinusoidal signal, to setting the frequency Ω_0 to 0. At the other extreme, the *highest* rate of variation possible for a DT signal is when it alternates signs at each time step, as in $(-1)^n$. A sinusoid with this property is obtained by taking $\Omega_0 = \pm\pi$, because $\cos(\pm\pi n) = (-1)^n$. Thus **all the action of interest with DT sinusoids happens in the frequency range** $[-\pi, \pi]$. Outside of this interval, everything repeats periodically in Ω_0, precisely because adding any integer multiple of 2π to Ω_0 does not change the value of the cosine in Equation (12.8).

It can be helpful to consider this DT sinusoid as derived from an underlying CT sinusoid $\cos(\omega_0 t + \theta_0)$ of period $2\pi/\omega_0$, by sampling it at times $t = nT$ that are integer multiples of some sampling interval T. Writing

$$\cos(\Omega_0 n + \theta_0) = \cos(\omega_0 nT + \theta_0)$$

then yields the relation $\Omega_0 = \omega_0 T$ (with the constraint $|\omega_0| \leq \pi/T$, to reflect $|\Omega_0| \leq \pi$). It is now natural to think of $2\pi/(\omega_0 T) = 2\pi/\Omega_0$ as the *period* of the DT sinusoid, measured in samples. However, $2\pi/\Omega_0$ may not be an integer!

Nevertheless, if $2\pi/\Omega_0 = P/Q$ for some integers P and Q, i.e., if $2\pi/\Omega_0$ is *rational*, then indeed $x[n+P] = x[n]$ for the signal in Equation (12.8), as you can verify quite easily. On the other hand, if $2\pi/\Omega_0$ is irrational, see Slide 12.12, the DT sequence in Equation (12.8) will not actually be periodic: there will be no integer P such that $x[n+P] = x[n]$ for all n.

With all that said, it turns out that the response of an LTI system to a sinusoid of the form in Equation (12.8) is a sinusoid of the same (angular) frequency Ω_0, whether or not the sinusoid is periodic. The easiest way to demonstrate this fact is to rewrite sinusoids in terms of **complex exponentials**.

■ 12.2.2 Complex Exponentials

The relation between complex exponentials and sinusoids is captured by Euler's identity, Slide 12.14:

$$e^{j\phi} = \cos\phi + j\sin\phi\ . \tag{12.9}$$

where $j = \sqrt{-1}$. This represents a complex number (or a point in the complex plane) that has a real component of $\cos\phi$ and an imaginary component of $\sin\phi$. It therefore has magnitude 1 (because $\cos^2\phi + \sin^2\phi = 1$), and makes an angle of ϕ with the positive real axis. In other words, $e^{j\phi}$ is the point on the unit circle in the complex plane (i.e., at radius 1 from the origin) and at an angle of ϕ.

A quick refresher on complex numbers: The complex number $c = a + jb$ can be thought of as the point (a, b) in the plane, and accordingly has *magnitude* $|c| = \sqrt{a^2 + b^2}$ and *angle* with the positive real axis of $\angle c = \arctan(b/a)$. Note that $a = |c|\cos(\angle c)$ and $b = |c|\sin(\angle c)$. Hence, in view of Euler's identity, we can also write the complex number in so-called *polar* form, $c = |c|.e^{j\angle c}$; this represents a point at distance $|c|$ from the origin, at an angle of $\angle c$.

The extra thing you can do with complex numbers, which you cannot do with just points in the plane, is *multiply* them. And the polar representation shows that the product of two complex numbers c_1 and c_2 is

$$c_1.c_2 = |c_1|.e^{j\angle c_1}.|c_2|.e^{j\angle c_2} = |c_1|.|c_2|.e^{j(\angle c_1 + \angle c_2)}\ ,$$

i.e., the magnitude of the product is the product of the individual magnitudes, and the angle of the product is the *sum* of the individual angles. It also follows that the *inverse* of a complex number c has magnitude $1/|c|$ and angle $-\angle c$.

Several other identities follow from Euler's identity above. Most importantly,

$$\cos\phi = \frac{1}{2}\Big(e^{j\phi} + e^{-j\phi}\Big) \qquad \sin\phi = \frac{1}{2j}\Big(e^{j\phi} - e^{-j\phi}\Big) = \frac{j}{2}\Big(e^{-j\phi} - e^{j\phi}\Big)\ . \tag{12.10}$$

Also, writing

$$e^{jA}e^{jB} = e^{j(A+B)}\ ,$$

and then using Euler's identity to rewrite all three of these complex exponentials, and finally multiplying out the left hand side, generates various useful identities, of which we only list two:

$$\begin{aligned} \cos(A)\cos(B) &= \tfrac{1}{2}\Big(\cos(A+B)+\cos(A-B)\Big) \; ; \\ \cos(A\mp B) &= \cos(A)\cos(B) \pm \sin(A)\sin(B) \; . \end{aligned} \tag{12.11}$$

12.2.3 Frequency Response

We are now in a position to determine what an LTI system does to a sinusoidal input. The streamlined approach to this analysis involves considering a *complex* input of the form $x[n] = e^{j(\Omega_0 n+\theta_0)}$ rather than $x[n] = \cos(\Omega_0 n + \theta_0)$. The reasoning and mathematical calculations associated with convolution work as well for complex signals as they do for real signals, but the complex exponential turns out to be somewhat easier to work with (once you are comfortable working with complex numbers!)—and the results for the real sinusoidal signals we are interested in can then be extracted using identities such as those in Equation (12.10).

It may be helpful, however, to first just plough in and do the computations directly, substituting the real sinusoidal $x[n]$ from Equation (12.8) into the convolution expression in Equation (12.4), and making use of Equation (12.11). The purpose of doing this is to (i) convince you that it can be done entirely with calculations involving real signals; and (ii) help you appreciate the efficiency of the calculations with complex exponentials when we get to them.

The direct approach mentioned above yields

$$\begin{aligned} y[n] &= \sum_{m=-\infty}^{\infty} h[m]\cos\Big(\Omega_0(n-m)+\theta_0\Big) \\ &= \Big(\sum_{m=-\infty}^{\infty} h[m]\cos(\Omega_0 m)\Big)\cos(\Omega_0 n+\theta_0) \\ &\quad +\Big(\sum_{m=-\infty}^{\infty} h[m]\sin(\Omega_0 m)\Big)\sin(\Omega_0 n+\theta_0) \\ &= \mathcal{C}(\Omega_0)\cos(\Omega_0 n+\theta_0)+\mathcal{S}(\Omega_0)\sin(\Omega_0 n+\theta_0) \; , \end{aligned} \tag{12.12}$$

where we have introduced the notation

$$\mathcal{C}(\Omega) = \sum_{m=-\infty}^{\infty} h[m]\cos(\Omega m) \; , \qquad \mathcal{S}(\Omega) = \sum_{m=-\infty}^{\infty} h[m]\sin(\Omega m) \; . \tag{12.13}$$

Now define the complex quantity

$$H(\Omega) = \mathcal{C}(\Omega) - j\mathcal{S}(\Omega) = |H(\Omega)|.\exp\{j\angle H(\Omega)\} \; , \tag{12.14}$$

which we will call the **frequency response** of the system, for a reason that will emerge immediately below. Then the result in Equation (12.12) can be rewritten, using the second identity in Equation (12.11), as

$$\begin{aligned} y[n] &= |H(\Omega_0)|.\Big[\cos\angle H(\Omega_0).\cos(\Omega_0 n + \theta_0) - \sin\angle H(\Omega_0)\sin(\Omega_0 n + \theta_0)\Big] \\ &= |H(\Omega_0)|.\cos\Big(\Omega_0 n + \theta_0 + \angle H(\Omega_0)\Big) \ . \end{aligned} \tag{12.15}$$

The result in Equation (12.15) is *fundamental and important*! It states that the entire effect of an LTI system on a sinusoidal input at frequency Ω_0 can be deduced from the (complex) frequency response evaluated at the frequency Ω_0. The amplitude or magnitude of the sinusoidal input gets scaled by the magnitude of the frequency response at the input frequency, and the phase gets augmented by the angle or phase of the frequency response at this frequency.

Now consider the same calculation with complex exponentials, Slide 12.16. Suppose

$$x[n] = A_0 e^{j(\Omega_0 n + \theta_0)} \quad \text{for all } n \ . \tag{12.16}$$

The convolution expression in Equation (12.4) then yields

$$\begin{aligned} y[n] &= \sum_{m=-\infty}^{\infty} h[m] A_0 e^{j\Big(\Omega_0(n-m)+\theta_0\Big)} \\ &= \Big(\sum_{m=-\infty}^{\infty} h[m] e^{-j\Omega_0 m}\Big) A_0 e^{j(\Omega_0 n + \theta_0)} \ . \end{aligned} \tag{12.17}$$

Thus the output of the system, when the input is the (everlasting) exponential in Equation (12.16), is the same exponential, except multiplied by the following quantity evaluated at $\Omega = \Omega_0$:

$$\sum_{m=-\infty}^{\infty} h[m] e^{-j\Omega m} = C(\Omega) - jS(\Omega) = H(j\Omega) \ . \tag{12.18}$$

The first equality above comes from using Euler's equality to write $e^{-j\Omega m} = \cos(\Omega m) - j\sin(\Omega m)$, and then using the definitions in Equation (12.13). The second equality is simply the result of recognizing the frequency response from the definition in Equation (12.14).

To now determine was happens to a sinusoidal input of the form in Equation (12.8), use Equation (12.10) to rewrite it as

$$A_0 \cos(\Omega_0 n + \theta_0) = \frac{A_0}{2}\Big(e^{j(\Omega_0 n + \theta_0)} + e^{-j(\Omega_0 n + \theta_0)}\Big) \ ,$$

and then superpose the responses to the individual exponentials, using the result in Equation (12.17). The result (after algebraic simplification) will again be the expression in Equation (12.15), except scaled now by an additional A_0, because we scaled our input by this additional factor in the current derivation (just to kick things up one step in generality).