Module 1, B-set 3: The Power of LTI

I'm feeling impulsive today...

September 16, 2016

Learning Goals

By the end of this building block, you should be able to...

- Use convolutions to implement filters and simulate effects of LTI systems
- Move things around in frequency space such that multiple users can share a medium

WARNING: By the time you're 4 or 5 hours into this, you should feel confident that you can complete the B-set in another 4-5 hours. If not, this is when you should **ask for help.** This means **talk to a colleague**, or **talk to a ninja**, or **track down an instructor**, or **send an email to an instructor**.

Input-output relationship of LTI systems (Estimated Time 1hr)

Being able to predict the output of a system given its input is tremendously useful in designing systems. The input-output relationship of LTI systems is relatively easy to compute, and LTI systems are good models for many real-world systems.

The output of a system when the input is $\delta[n]$ is denoted by $h[n]$, and is called the **impulse response** of the system. If we can decompose an input signal into a sum of scaled, shifted impulses, the output will be a sum of scaled shifted impulse responses.

1. Calculate the impulse response of a system that calculates the three element trailing average of the input:

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$$

   In other words, what does the output of the system look like if the input is $x[n] = \delta[n]$ – i.e., a $x = 1$ at $n = 0$ and 0 everywhere else?

Once you have the system's impulse response, it's easy to calculate the response of the system to a given input: simply decompose the input into weighted impulses, ask what the response is to each impulse, and then add those responses together.

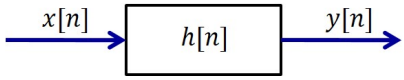


Figure 1: Input-output Block diagram of DT system.

2. Try this. As $x[n]$, use the signal given in Figure 2. For your system, use a three element trailing average filter as described above. Using the idea of impulse response and decomposing the input into impulses, calculate the output $y[n]$. Do so graphically – i.e., draw pictures of stem plots.

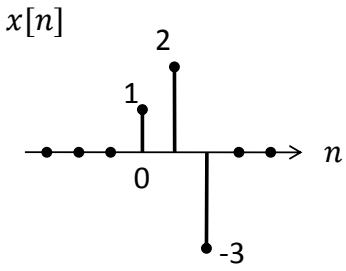


Figure 2: Example DT Signal.

Impulse responses and convolution (Estimated Time 2hrs)

We can do the same for more generic signals, and develop a mathematical expression relating the input and the output of the system for a general signal.

Consider the signal $x[n]$ shown in Figure 3, where the at time n, $x[n]$ has the value q_n. $x[n]$ is the sum of the scaled, shifted impulses shown in the second column of the figure.

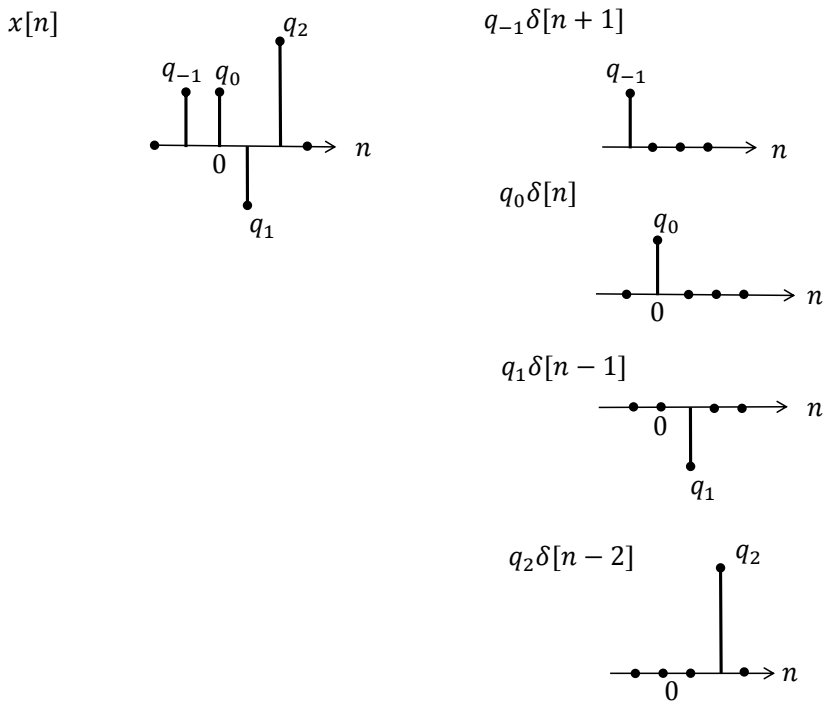


Figure 3: Decomposition of a DT Signal.

We can write

$$x[n] = q_{-1}\delta[n+1] + q_0\delta[n] + q_1\delta[n-1] + q_2\delta[n-2].$$

We can also write this in summation form as

$$x[n] = \sum_{k=-1}^{2} q_k\delta[n-k] \tag{1}$$

But because $q_k = x[k]$, we may write the above expression as

$$x[n] = \sum_{k=-1}^{2} x[k]\delta[n-k]\,. \tag{2}$$

This seems very self-referential, but the purpose of writing it this way is to derive a general property of a system. You should view this as a technique to arrive at a useful result at the end.

If this summation is extended from $-\infty$ to $+\infty$ we can handle any length signal, and signals defined for a finite duration can be extended by using zeros.

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]\,. \tag{3}$$

If we stick the signal above through an LTI system, each scaled, shifted impulse in the summation will result in a scaled shifted impulse response in the output of the system.

Therefore

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

This is called the **convolution** of $x[n]$ and $h[n]$, and is also denoted using the * sign, i.e.

$$y[n] = x * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

3. Consider a simple abstraction of a wired communication system (e.g. optical fiber, ethernet, cable modem) depicted in Figure 4. The system is modeled as an LTI system with input signal $x[n]$ and impulse response $h[n]$. The signal $x[n]$ is made up of a sequence of ± 1 values that are spread apart in time, with zeroes in between. In this case successive ± 1 values are separated by 7 samples. A +1 value represents a "1" bit, and a -1 value represents a "0" bit. In this example 1 bit of data is transmitted every 7 samples. Given a particular sampling rate, this can be converted into a data rate expressed in bits/second.

   (a) For the given example, sketch the output $y[n]$.

(b) What potential problems can you anticipate if the time between the ± 1 samples is too short?

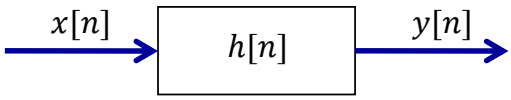
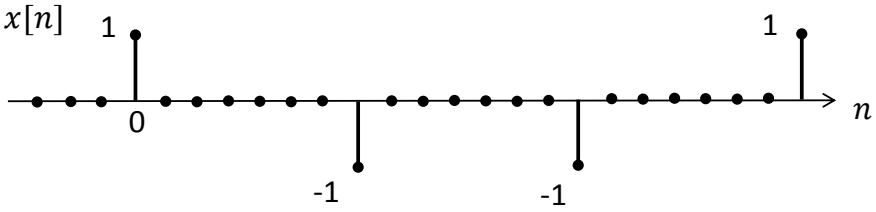
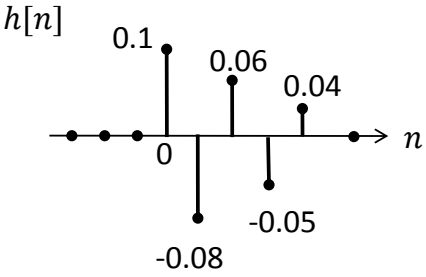


Figure 4: An LTI system.





4. In this problem, you are going to simulate what it would sound like if Slash (a guitarist who was popular about the time when you were born) were playing his guitar in a stairwell at Olin.

   We are going to model the stairwell as an LTI system with the input being the sound that is played, and the output being the sound that gets to your ear/microphone in your computer. Figure 5 shows a block diagram for this system.

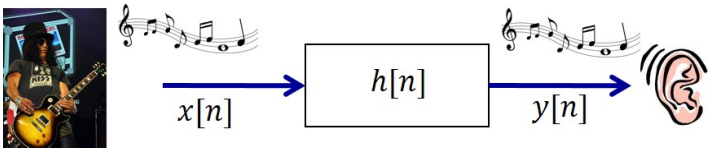


Figure 5: Block diagram of music being played in a stairwell.

   To do this, you will need an estimate of the impulse response in a stairwell at Olin. One of the instructors went to a stairwell at Olin, snapped their fingers and recorded the sound.

   (a) Explain why the recording of a finger being snapped is a decent approximation to the impulse response of this system

   (b) At the MATLAB Prompt, type

```
load guitar_stairwell
```

   You should see three variables in your workspace `h_stairwell x Fs`. `Fs` is the sampling frequency, `x` is an audio signal of a guitar riff, and `h_stairwell` is an estimate of the impulse response measured at the stairwell.

   Using what you know about the convolution and impulse responses simulate the effect of the guitar sample being played in a stairwell. You will find MATLAB's `conv` function useful here. Please turn in any code you used here.

Frequency Response and Convolution (Estimated Time 2 hours)

The convolution tells us the relationship between the input and ouput signals in the time domain. Here, we are going to look at that relationship in the frequency domain, and observe that the frequency response of a DT LTI system is the DTFT of its impulse response.

Now you might have read the preceding paragraph and said "WTF does that mean, and why should I care?" Great question.

The key idea here is that *by measuring the impulse response of the system, you can also obtain the frequency response of the system.* This is kind of amazing at first glance: by snapping my fingers in the stairwell, I have measured how the stairwell will react to ANY frequency.

Why is this possible? Well, there's a question for you to ponder...

From Impulse Response to Frequency Response

Here's the math behind the claim above. Consider an LTI system with an input $x[n]$ and output $y[n]$, as shown in Figure 4.

The input and output are related by the convolution

$$y[n] = x * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{4}$$

If we take the DTFT of the expression above and do a change of variables, we will find that

$$Y(\Omega) = H(\Omega)X(\Omega)\,, \tag{5}$$

where $Y(\Omega)$ is the DTFT of $y[n]$, $X(\Omega)$ is the DTFT of $x[n]$ and $H(\Omega)$ is the DTFT of $h[n]$.

$H(\Omega)$ is the frequency response of this system, and it tells us how the system effects the frequencies contained in $x[n]$. Therefore, the DTFT of the impulse response of a system is the frequency response of the system.

Equations (5) and (4) show the relationship between the input and output of LTI systems in the frequency and time domains respectively. This relationship can be captured in Figure 6.

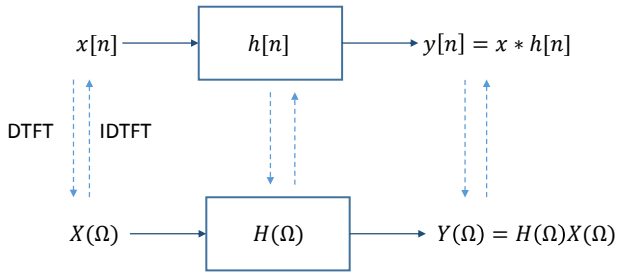


Figure 6: LTI Input-Output Relationship in Time and Frequency.

Note that this is another example of going from the time domain to the frequency domain and back. If you know the impulse response, you can figure out the frequency response, and vice-versa.

Ideal Low-pass Filter

AS an application, let's think about the frequency response of an ideal low-pass filter. What we're going to do here is kind of the reverse of the notch filter: in the notch case, you were given the difference equation, and by transforming it to the frequency domain, you were able to understand its behavior. Here we're going to start with a desired behavior, and use that behavior to come up with the impulse response (effectively the difference equation) from that behavior.

$H(\Omega)$ of our ideal filter shown in Figure 7. The filter has a cutoff frequency of Ω_c radians/sample.

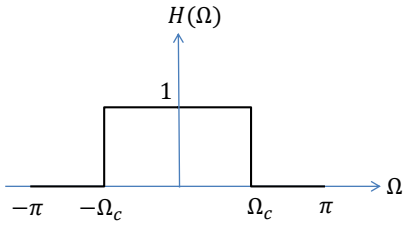


Figure 7: Frequency response of an ideal low-pass filter.

The impulse response corresponding to this filter can be obtained using the IDFT equation, taking an integral and simplifying to get

$$\begin{aligned} h[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\Omega)e^{j\Omega n}\,d\Omega \\ &= \frac{\sin(\Omega_c n)}{n\pi} \end{aligned} \tag{6}$$

This function is illustrated in Figure 8, which illustrates the case $\Omega_c = \frac{\pi}{8}$. Functions of this form are called `sinc` functions. There is no commonly accepted definition of the sinc function, but in general it has $\frac{\sin x}{x}$ form. In MATLAB, the sinc function does the following

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \tag{7}$$

5. In MATLAB type `load handel` to load an audio signal, which is contained in a vector `y`.

   (a) Your job is to low-pass filter this signal by convolving it with an appropriate sinc function. The low-pass filter should have a cutoff frequency of $\Omega_c = \frac{\pi}{4}$ radians/sample. The following code segment can help you create a sinc function corresponding to a LPF with this cutoff frequency.

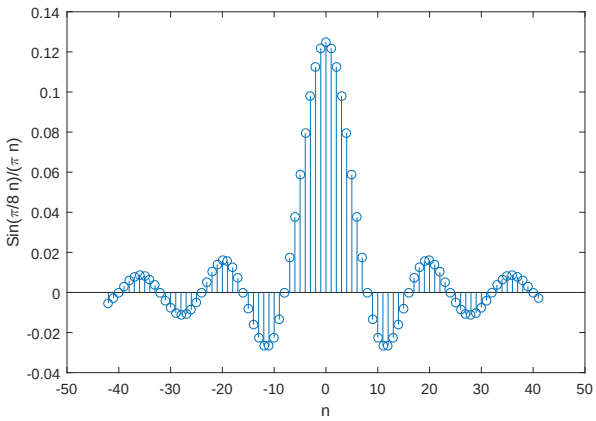


Figure 8: Sinc function.

```
% make a vector for the time indices
% technically, this is suppsed to go from
% - infinity to +infinity, but this will be long enough
n = [-42:41];
% generate a sinc function corresponding to
% a LPF with cutoff frequency of wc
h = wc/pi*sinc(wc*n/pi);
```

On the same axes, plot the magnitude of the fft of the signal before and after the filtering.

(b) Next, using the convolution operation, high pass filter your signal with a cut off frequency of $\Omega_c = \frac{\pi}{8}$. The impulse response of a high pass filter is with cutoff frequency Ω_c is

$$h[n] = \delta[n] - \frac{\sin(\Omega_c n)}{n\pi} \tag{8}$$

Plot the magnitude of the fft of the signal before and after the filtering.

More applications to communications (Estimated Time 1 hr)

6. In class and the last BSet, you saw how multiplying a time-domain signal by a moves things around in frequency. Besides enabling signals to propagate through bandpass channels, this approach is also used to accomodate multiple users in the same channel, by placing their signals at different locations in frequency space. This is done in radio stations for instance.

   Suppose we have signals $x[n]$ and $w[n]$ whose DTFT's are shown in Figure 9.

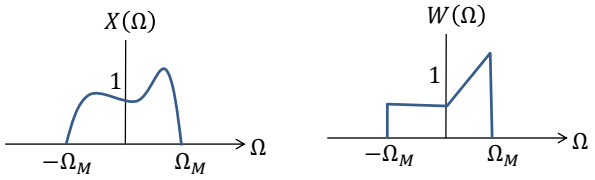


Figure 9: Example baseband signals

   Consider the signal

$$y[n] = \cos(\Omega_x n)x[n] + \cos(\Omega_w n)w[n] \tag{9}$$

   $y[n]$ serves as a DT model for the received signal by your radio, if there are two radio stations present, transmitting at two different frequencies.

   Suppose that $\Omega_w - \Omega_x > 2\Omega_M$, and $\Omega_x \gg \Omega_M$ and $\Omega_w \gg \Omega_M$.

(a) Use the identity $\cos\theta = \frac{1}{2}e^{j\theta} + \frac{1}{2}e^{-j\theta}$, and the frequency-shift property of the DTFT to sketch $Y(\Omega)$.

(b) Describe how you may recover $x[n]$ and $w[n]$ from $y[n]$.

(c) Load the file 'TwoAM.wav' using MATLAB's audioread function. DO NOT PLAY THIS SOUND DIRECTLY, IT HAS HIGH FREQUENCIES. This file contains the signal $y[n]$ which is the sum of two signals $x[n]$ and $w[n]$ which have been multiplied by cosines. The frequencies of the cosines in radians/sample are $\Omega_x = (16000/Fs)\pi$ and $\Omega_w = (6000/Fs)\pi$. For these examples, you can assume that $\Omega_M = 0.3$ radians/sample. Plot the magnitude of the FFT of this signal to visualize it.

(d) In MATLAB, write code to recover $w[n]$ and $x[n]$ from $y[n]$ Please verify that your system works by playing the two sounds. One should be a violin sample and another a guitar. Please turn in your code.