Signal and Information Processing

Alejandro Ribeiro

April 5, 2016

Contents

Chapter 1

Linear Systems

1.1 Linear Time Invariant (LTI) Systems

When we implement Fourier transform analysis, it allows us to enter the frequency domain where it is easier to discern patterns and properties relevant to signal and information processing. In addition, we are able to analyze and design linear time invariant (LTI) systems. Here we will give a brief introduction to this topic.

Before defining what a system is, we will first discuss two key properties observed when analyzing LTIs. When given a delta function as an input, we can use the impulse response to characterize what will occur to the system when given any signal as an input. Furthermore, given any input signal, we can characterize the output response as a convolution with the delta impulse response. This gives equivalent properties in the frequency domain, ie $\mathcal{F}$(impulse response), in which the output spectrum is the multiplication of the input spectrum with the frequency response.

If we consider discrete time systems, as in Figure 1.1, a system is characterized by the relationship between an input $x(n)$ and its output $y(n)$. This relationship is between functions, and not values. Therefore, each output value $y(n)$ depends on all input values $x(n)$. In other words, $y(0)$ is not solely dependent on $x(0)$, but on all input values $x(n)$, and so forth. This is the same for if we considered continuous time systems. We will further define a system as time invariant if a delayed input yields a delayed output, ie if input $x(n)$ yields output $y(n)$, then input $x(n-k)$ yields $y(n-k)$. We can simply consider

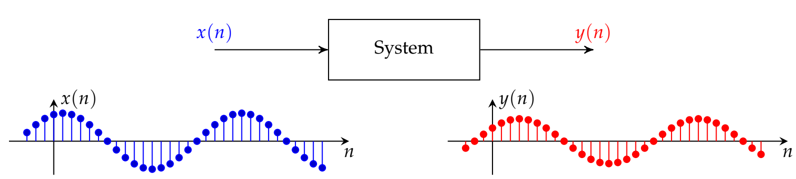

Figure 1.1. Given a signal $x(n)$ as an input to a linear system, the output $y(n)$ depends on all input values of $x(n)$.

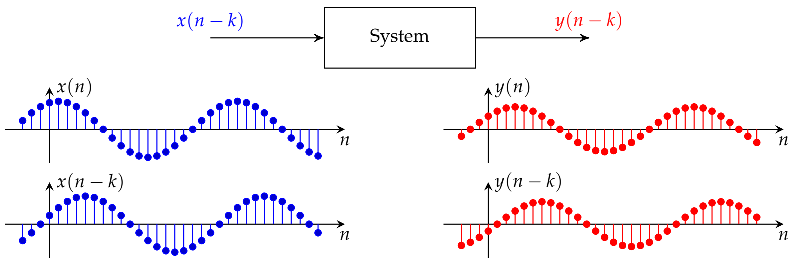

Figure 1.2. Given a signal $x(n)$, the signal $x(n-k)$ is an input to a linear system k time units later, and thus the output is $y(n-k)$.

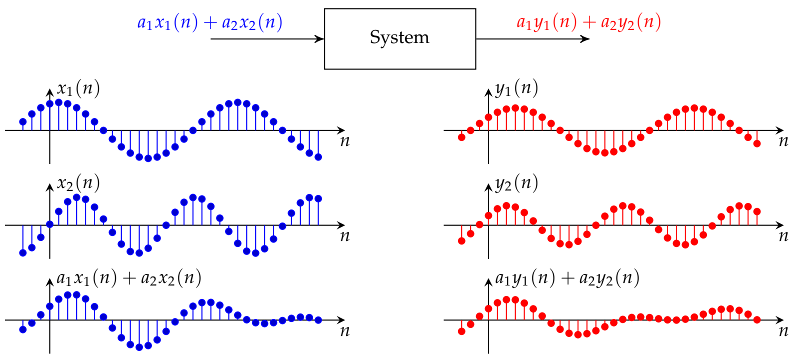

Figure 1.3. Given an input $a_1x_1(n) + a_2x_2(n)$, the linear combination of a signal $x_1(n)$ and $x_2(n)$, the output will be the linear combination $a_1y_1(n) + a_2y_2(n)$ of the respective outputs, $y_1(n)$ and $y_2(n)$.

this as applying an input k time units later, gets an output k time units later, as seen in Figure 1.2. Lastly, we will consider a system to be linear if given an input of a linear combination of inputs, the output is also the same linear combination of those respective outputs, as seen in Figure 1.3. For example, given input $x_1(n)$, which yields output $y_1(n)$ and a second input $x_2(n)$, which yields $y_2(n)$, an input of the linear combination $a_1x_1(n) + a_2x_2(n)$ yields an output $a_1y_1(n) + a_2y_2(n)$, an equivalent linear combination.

A linear time invariant system is also considered to be a LTI filter, or simply, a filter. As mentioned, the impulse response to a LTI system is the output when the input is a delta function. Mathematically, given an input $x(n) = \delta(n)$, the output is $y(n) = h(n)$, where $h(n)$ is defined as the impulse response, given in Figure 1.4. As we are considering a time invariant system, a shift of the delta function, $\delta(n-k)$, will create a shift in the impulse response, $h(n-k)$, as in Figure 1.5. Additionally, since the system is linear, scaling the input by a signal $x(k)$, $x(k)\delta(n-k)$, will cause a scaling of the impulse response, $x(k)h(n-k)$, as in Figure 1.6. Considering a linear combination of these properties, we can then see that given an input $x(k_1)\delta(n-k_1) + x(k_2)\delta(n-k_2)$, the output response can be modeled by $x(k_1)h(n-k_1) + x(k_2)h(n-k_2)$, as in Figure 1.7. We can see from the combination of shifting, scaling and summing, that this is a Convolution. If given a signal x, we can see that it can be written as in equation 1.1, since for a delta function $\delta(n-k)$, all values

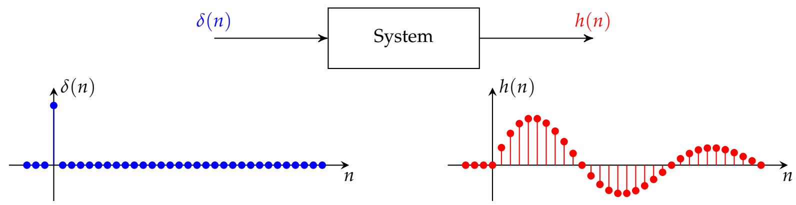

Figure 1.4. Given an input $x(n) = \delta(n)$, the output is the impulse response $y(n) = h(n)$.

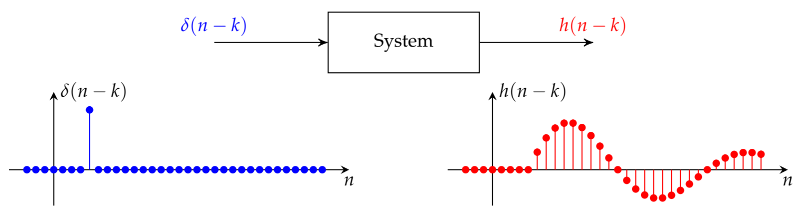

Figure 1.5. Given an input of a shifted delta signal $x(n) = \delta(n-k)$, the output is the shifted impulse response $y(n) = h(n-k)$.

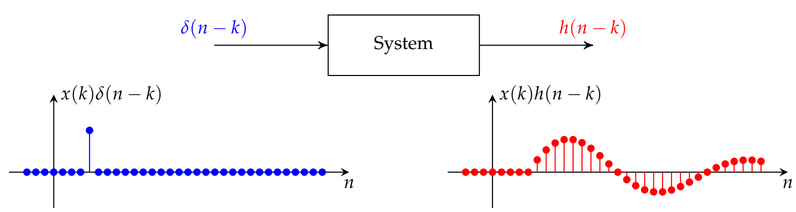

Figure 1.6. Given an input of a scaled, shifted delta signal $x(n) = x(k)\delta(n-k)$, the output is the scaled, shifted impulse response $y(n) = x(k)h(n-k)$.

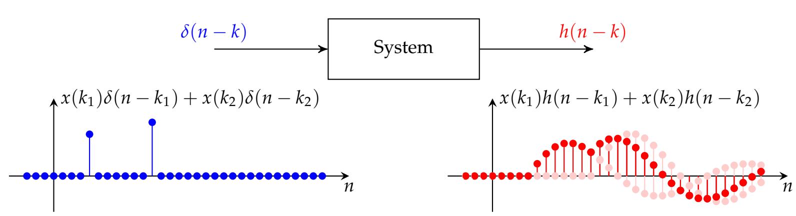

Figure 1.7. Given a linear combination of an input of a scaled, and shifted delta $x(n) = x(k_1)\delta(n-k_1) + x(k_2)\delta(n-k_2)$, the output is the linear combination of the scaled and shifted impulse response $y(n) = x(k_1)h(n-k_1) + x(k_2)h(n-k_2)$.

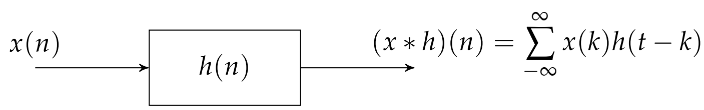

Figure 1.8. Given a signal x and its impulse responce h, we can see that the result is the convolution $x * h$

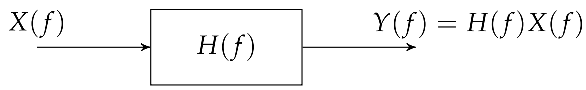

Figure 1.9. Given a signal in the frequency domain X and its impulse responce H, we can see that the result is the multiplication HX

where $n \neq k$ are 0, and so when $k = n$ is the only term in the delta function with a value of 1. Thus, we are only retaining the value in the summation were $k = n$, and such is equivalent to $x(n)$. Using this expression, it becomes clear that we have a shift, scale and summation, and therefore a convolution.

$$x(n) = \sum_{k=-\infty}^{+\infty} x(k)\delta(n-k) \tag{1.1}$$

Using this expression, we can say that given $x(n)$ as the input signal, we will have an output of the LTI system with the impulse response h, in equation 1.2. Using our definition of convolution, we can see that this response is the convolution of x and h, giving $y = x * h$.

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n-k) \tag{1.2}$$

We can formally write this information using the theorem:

Theorem 1 *A linear time invariant system is completely determined by its impulse response h. In particular, the response to input x is the signal $y = x * h$.*

Figure 1.8 gives an example of such an LTI system. It is important to note that the restrictions of linearity and time invariance induce a very strong structure on the signal. As previously determined, it is typically easier and more useful to analyze signals in the frequency domain. We already discussed that given and impulse response h, we can define its associated frequency response H by taking the Fourier Transform of the impulse response, given by $H = \mathcal{F}(h)$. Thus, we can define the corollary to our theorem above as:

Corollary 1 *A linear time invariant system is completely determined by its frequency response H. In particular, the response to input X is the signal $Y = HX$.*

Refer to Figure 1.9 for an explicit illustration. This is true for any LTI system, which makes it very useful for designing a system in the frequency domain and then implementing in time. We will define now a causal filter, which is such a filter with $h(n) = 0$ for all $n < 0$. This is intuitive, because it is not possible to respond to a spike in time before seeing that

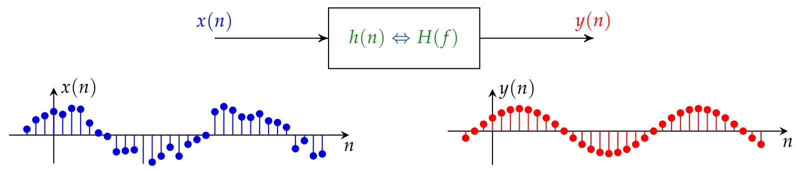

Figure 1.10. Given a signal $x(n)$ we implement a filter of the impulse response h to get the output $y(n)$.

spike. Additionally, to clarify that the output response $y(n)$ is only affected by past inputs of $x(k)$, we define $k \leq n$, as in equation 1.3.

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n-k) = \sum_{k=-\infty}^{n} x(k)h(n-k) \tag{1.3}$$

If we are given a filter that is not causal, but has $h(n) = 0$ for all $n < N$, we can cause the filter to be causal by introducing a delay to the system, given by $\tilde{h}(n) = h(n-N)$. Defining this in the frequency domain, we have the frequency response of the delayed filter as $\tilde{H}(f) = H(f)e^{j2\pi fN}$, equivalent to the delayed filter $\tilde{h}(n)$ in the time domain. A second type of filter, a causal finite impulse response filter (FIR) is defined as one for which $h(n) = 0$ for all $n \geq N$. This filter is of length N, since for all n greater than N, the value of $h(n)$ is null. Explicitly, we can write the output of the filter response $y(n)$ at time point n as in equation 1.4. From this equation, we can see that the input signal $x(n)$ can be defined as a vector, $x_N(n) = [x(n); x(n-1); \ldots; x(n-N+1)]$ and the impulse response h can be defined as the vector $h = [h(0), h(1), \ldots, h(N-1)]$. Using these definitions, we can see that the output at time n is given by 1.5.

$$y(n) = h(0)x(n) + h(1)x(n-1) + \ldots h(N-1)x(n-N+1) \tag{1.4}$$

$$y(n) = h^T x_N \tag{1.5}$$

1.2 Linear Time Invariant (LTI) Systems

A useful application of LTI systems is to design and implement a filter to remove noise from (or smooth) a signal $x(n)$, as seen in Figure 1.10. We know that all LTIs are completely determined by their impulse responses h. Thus, we can design an impulse response h and implement a filter as the convolution in time, $y = x * h$. Additionally, we know by the corollary, that all LTIs are completely determined by their frequency responses H, and so we can also design H and implement a filter as the spectral product $Y = HX$. We know that both time and frequency representations are equivalent. Therefore, we can find an optimal path in implementing and designing a filter, as seen in Figure 1.11. We can identify pattern transformations in the frequency domain, given by designing H. We can then use the inverse DTFT to compute the impulse response, given by $h = \mathcal{F}^{-1}(H)$. Lastly, we can implement a convolution in time, $y(n) = (x * h)(n)$. Unfortunately, an impulse response $h = \mathcal{F}^{-1}(H)$ is typically not causal or finite. We must

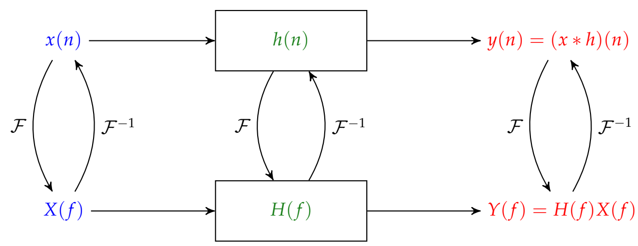

Figure 1.11. Illustrates that we can switch between the time and frequency domain for equivalent representations of the signal $x(n)$, the impulse response $h(n)$ and the output response $y(n)$.

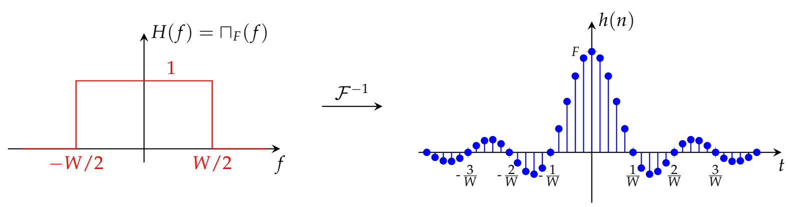

Figure 1.12. The time representation of a low pass filter $H(f)$ with cutoff frequency $W/2$ is the sinc function $h(n)$ which is both infinite and non-causal.

implement a low pass filter with cutoff frequency $W/2$, given by $H(f) = \sqcap_W(f)$. We know that this filter is given by a sinc function, as in equation 1.6, seen in Figure 1.12.

$$h(n) = \int_{-f_s/2}^{f_s/2} H(f) e^{j2\pi f n T_s}\, df = W \text{sinc}(\pi W n T_s) \tag{1.6}$$

However, a sinc function is an infinite filter. Additionally, it is an impulse response centered at time $n = 0$, and so it is non-causal. Therefore, we must multiply by a window to create a finite response with N nonzero coefficients, as in Figure 1.13. This transformation of $h(n)$ into a finite impulse response is given by $h_w(n) = h(n)w(n)$, where the window $w(n) = 0$ for all $n \notin [N_{\text{min}}, N_{\text{max}}]$. This gives us a filter length of $N = N_{\text{max}} - N_{\text{min}} + 1$. Additionally, we must delay the impulse response $h(n)$ so that we have a causal filter with $h(n) = 0$ for all $n \leq 0$, as in Figure 1.14. This is a shift of the finite impulse response $h_w(n)$, giving us $h_w(n - N_{\text{min}})$. We want to choose borders N_{min} and N_{max} such that we retain the highest values of $h(n)$. Often, this is around $n = 0$, but this is not always the case. We know that a multiplication in time domain is a Convolution in the frequency domain. As a result, instead of filtering with $H(f)$, we filter with $H_w = H * W$. When designing this window W, we thus want to design a window that doesn't introduce much distortion to the signal, and thus we wish to choose windows with spectrum $W = \mathcal{F}(w)$ close to the delta function. When introducing the time delay, this is a multiplication with a complex exponential in the frequency domain, giving us $H_w(f)e^{j2\pi f N_{\text{min}} T_s}$.

In summary, the procedure to design time coefficients of a FIR filter are as follows:

1. Spectral analysis to determine filter frequency response $H(f)$
2. Inverse DFT (not DTFT) to determine impulse response $h(n)$

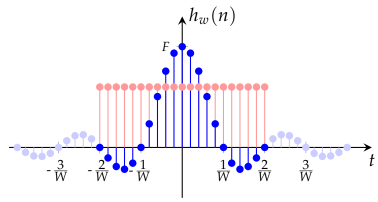

Figure 1.13. Transform impulse response $h(n)$ into a finite response by chopping with a window $w(n)$ such that $h_w(n) = h(n)w(n)$.

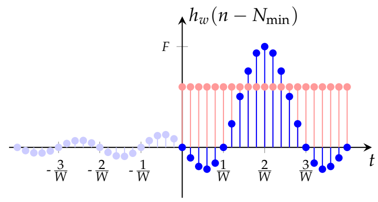

Figure 1.14. Transform $h_w(n)$ into a causal response by shifting the signal $h_w(n - N_{\min})$.

3. Determine number of coefficients N and coefficient range $[N_{\min}, N_{\max}]$
4. Select window $w(n)$ such that it alters the spectrum to $H_w = H * W$
5. Shift impulse response by $N_{\min}$ time steps to make filter causal

We know that the output $y(n)$ of the FIR filter is given by the convolution value in equation 1.7. Since h is finite and causal, there are only N nonzero terms. Therefore, we can make k go from 0 to $N-1$, given in equation 1.8. This can be easier to visualize when written in expanded form, such that $y(n) = h(0)\,x(n) + h(1)\,x(n-1) + \ldots + h(N-1)\,x(n-N+1)$. This expression is the inner product of two vectors, and can be implemented with a shift register.

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{1.7}$$

$$y(n) = \sum_{k=n-(N-1)}^{n} x(k)h(n-k) = \sum_{l=0}^{N-1} h(l)x(n-l) \tag{1.8}$$

A shift register is implemented using a row of delay operations, of which given a signal $x(n)$ the output $y(n)$ is computed through the shift of the signal x, product of the units to multiply with filter coefficients $x(n)$, and summation of the units to aggregate the products $h(k)x(n-k)$. See Figure 1.15 for an illustration. Shift registers are used in hardware and occasionally software. An implementation of using a filter is in voice recognition. We can compare a given word to an average spectra $\bar{X}$ of the magnitude of the word to be recognized. This comparison is done using the inner product, $X^T\bar{X}$. This is equivalent to using $\bar{X}$ to filter X, where we get $Y(f) = H(f)X(f)$ with $H(f) = \bar{X}$. If we want to evaluate the impulse response $h(n)$, we can take the inverse DFT of the average spectra $\bar{X}$, as in Figure 1.16.

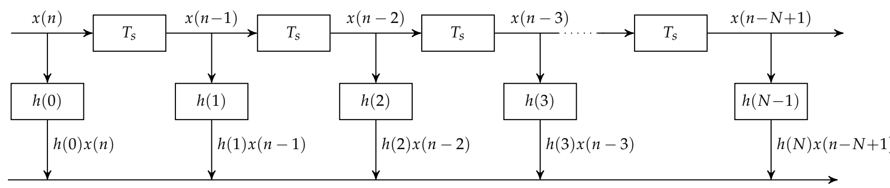

Figure 1.15. Implementation of a shift register.

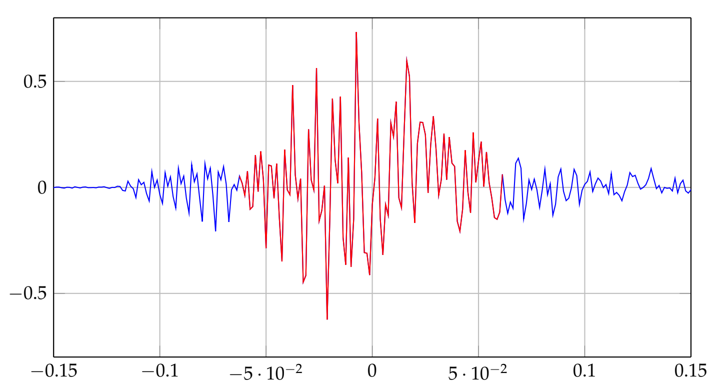

Figure 1.16. Impulse response $h(n)$, colored in red where window is applied to keep $N = 1,000$ largest consecutive taps.