A Brief Introduction to Digital Signal Processing

Lester Fan

March 31, 2018

1 Introduction

Welcome to my class at this year's offering of Columbia SPLASH! I am a Junior at Rutgers University pursuing a double major in Electrical Engineering and Computer Science and a minor in Math. These notes were made partly because I wanted to prepare for the class, and because I wanted you guys to have something to read over at home and play around with after the class if you were interested. I decided to put a lot of material here just to be complete, so I won't be following these notes exactly. This largely reflects the Signal Processing I have learned so far during my time at Rutgers, most of which I have learned from Professor Sarwate and Professor Bajwa in the ECE department. Everything I prepared for this class will be online at `https://github.com/lesterfan/Splash18`. I haven't put in much work to organize my code, but you can always contact me for help if you want to play around with some of the stuff I will go over today. If you have any questions at all, feel free to contact me at `hiiamlife@gmail.com`.

2 Math Review

We will begin with a review of some math that may be important for this class.

Definition 1. *The **imaginary constant** is the number j such that:*

$$j^2 = -1$$

and

$$j = \sqrt{-1}$$

Definition 2. *The set of **complex numbers** is defined as*

$$\mathbb{C} = \{a + bj, a \in \mathbb{R}, b \in \mathbb{R}\}$$

In other words, it is the set of all numbers that can be written in the form $a + bj$ for some real numbers a and b, where j is defined as above.

By using complex numbers, we can almost look at numbers as if they have two "dimensions", the real dimension and the "imaginary" dimension. Another natural way to look at complex numbers is in terms of a magnitude and an angle, as formalized below.

Theorem 1. *(**Euler's Identity**) For all $\theta \in \mathbb{R}$, we have:*

$$e^{j\theta} = \cos(\theta) + j\sin(\theta)$$

Theorem 2. *(**Polar Form of a Complex Number**) Every number $z = a + bj \in \mathbb{C}$ can be expressed in terms of $z = re^{j\theta}$ for $r \in \mathbb{R}$ called the **magnitude** of the complex number and $\theta \in \mathbb{R}$, called the **phase** of the complex number.*

Proof. Let $a + bj \in \mathbb{C}$. Define $\theta := \tan^{-1}(\frac{b}{a})$ and $r = \sqrt{a^2 + b^2}$. Then, we have $z = a + bi = re^{j\theta}$. □

Example 1.

$$j^j = (e^{j\frac{\pi}{2}})^j = e^{-\frac{\pi}{2}}$$

(who would have known it would be a real number!?).

Exercise 1. *Try finding the value of $\sqrt{j}$ by changing j from rectangular to polar form (and then back to rectangular again if you're ambitious).*

3 Properties of Signals

We will start by formally defining some properties of signals and build our way up to analyzing and manipulating them to do some pretty cool stuff!

Definition 3. *A **signal** is defined as any function that conveys information about the behavior or attributes of some phenomena.*

A signal can either be a *continuous time* signal, or a *discrete time* signal.

Definition 4. *A **continuous time** (CT) signal is a signal (or function) whose domain is a continuum, or an uncountable set. Typically, this means that its domain is the real numbers.*

Continuous time signals basically include every function that you learn about in your Algebra/Precalculus classes. You can view these are "regular" functions/signals, and examples of these include sound as a function of air pressure, a ball's position as a function of time, and so on.

Definition 5. *A **discrete time** (DT) signal is a signal (or function) whose domain is finite or countable. Typically, this means that its domain is the integers.*

These types of signals may be weird to think about at first, but these types of signals are extremely important because unlike continuous time signals, they can be stored in some way like a computer and reflect how we take measurements of things in the real world. Finally, I would like to introduce a special signal that you may not have seen before called the delta function.

Definition 6. *Call the DT signal*

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \textit{otherwise} \end{cases}$$

*the **Kronecker Delta Function**.*

Theorem 3. *Every DT signal $x[n]$ can be expressed as*

$$x[n] = \sum_{m--\infty}^{\infty} x[m]\delta[n-m]$$

Proof. (Rough idea) At a point m, $x[n]\delta[n-m] = x[m]$ (can you see why?). Now if we sum up this value for all m, we will have the original signal $x[n]$. □

4 Properties of (Linear and Time Invariant) Systems

Now that we know a little bit about signals, we would like to start talking about doing basic processing on such signals. To process a signal, we pass it through a **system**.

Definition 7. *A **system** H is an operator that takes a signal $x(t)$ or $x[n]$ as input and returns another signal $y(t)$ or $y[n]$ as output. Like signals, systems can be either continuous or discrete.*

From Algebra/Precalculus, we can already think of some ways we can modify signals, perhaps by scaling and shifting them. I won't include a review of basic scaling and shifting operations in these notes, but you can review them at `https://tinyurl.com/splashDSP` (and I will probably be going through them during the class as well).

Example 2. *The system*

$$H(x(t)) = 2x(t-5)$$

takes in as input a continuous time signal $x(t)$ and returns as output $y(t) = 2x(t-5)$, a copy of $x(t)$ that is scaled by 2 in amplitude and delayed by 5.

Usually, we are interested in real systems, so we will be studying mostly causal systems in this class.

Definition 8. *A system is **causal** if the current output only relies on past and present inputs (and outputs).*

The goal of Signal Processing is to design systems to perform specific tasks of our choosing. However, the definition of systems that we have right now is too general, and a bit too complicated for us to do any analysis with right now. Therefore, we will focus on systems that are linear and time invariant.

Definition 9. *A system is **linear** if for every scalar λ and every signal $x(t)$ and $y(t)$,*

$$H(x(t) + \lambda y(t)) = H(x(t)) + \lambda H(y(t))$$

Definition 10. *A system is **time invariant** if for every scalar t_0,*

$$H(x(t)) = y(t) \implies H(x(t-t_0)) = y(t-t_0)$$

Example 3. *The system*

$$H(x(t)) = 3x(2(t-2))$$

is both linear and time invariant, while the system

$$H(x(t)) = x(t)^2$$

is not linear but is time invariant. (These can be verified by checking the definitions).

The reason why we are so interested in LTI systems in this class (and in Signal Processing in general) is due to the following theorem.

Theorem 4. *The behavior of a discrete linear and time invariant (LTI) system can be completely characterized by the output when the input $x[n] = \delta[n]$ is sent into the system. The output signal*

$$h[n] = H(\delta[n])$$

is called the ***impulse response*** *of the system.*

Proof. Let $x[n]$ be a discrete time signal and H a linear time invariant system. Let $h[n]$ be the impulse response (defined above) of the system. From before, we know that we can always write

$$x[n] = \sum_{m=-\infty}^{\infty} x[m]\delta[n-m]$$

Now, we can find $H(x[n])$.

$$H(x[n]) = H(\sum_{m=-\infty}^{\infty} x[m]\delta[n-m])$$

But we know that H is a linear and time invariant system, so we can simplify:

$$H(x[n]) = \sum_{m=-\infty}^{\infty} x[m]H(\delta[n-m]) = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$$

Therefore, if we only have the impulse response of a LTI system h[n] and we wanted to find $H(x[n])$ for any arbitrary DT signal $x[n]$, we could find it by simply computing

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

so the impulse response $h[n]$ contains all of the information about the system that we need to know. □

Definition 11. *We define the sum introduced in the proof above as the* ***convolution*** *sum of two discrete signals $x[n]$ and $h[n]$:*

$$(x \star h)[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$$

Theorem 5. *Suppose a linear and time invariant (LTI) system H has impulse response $h[n]$. Then, we have that for every input signal $x[n]$, the output signal $y[n]$ is*

$$y[n] = H(x[n]) = (x \star h)[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$$

Example 4. *Suppose a system H has impulse response $h[n] = \{\underline{0}, 1, 0, 0, 1\}$, where the underlined term represents the value at $n = 0$. A signal $x[n] = \{\underline{1}, 1\}$ is sent through the system. Then, the output $y[n]$ is given by $\{\underline{0}, 1, 1, 0, 1, 1\}$.*

At this point, a quick demo involving violins, IceJJFish, and canyons will be given.

5 The Frequency Domain Interpretation of CT Signals

Now, we will begin to work our way to understanding signals enough to know what we want to do with them. One way of gaining more insight into a signal is to look at it in the frequency domain, which is similar to what our brain does every time we listen to music. This section will formalize the idea of how to look at a CT signal in terms of frequency.

Definition 12. *A CT signal over the field $F = \mathbb{R}$ which has a pure frequency f is of the form*

$$x(t) = Acos(2\pi ft + \phi) = Acos(\Omega t + \phi)$$

Call f the ***frequency*** *of $x(t)$ in Hertz or 1/second, and call $\Omega = 2\pi f$ the frequency of $x(t)$ in units radians/second. Call ϕ the* ***phase shift****. Radians/second is mostly used for signal processing applications, while Hertz is used for communications.*

Definition 13. *A CT signal over the field $F = \mathbb{C}$ which has a pure frequency f is of the form*

$$x(t) = Ae^{j2\pi ft+\phi} = Ae^{j\Omega t+\phi}$$

Joseph Fourier came up with the result that every periodic signal can be expressed as a linear combination of these signals with pure frequency, the idea of a Fourier Series representation as formalized below.

Theorem 6. *(**Continuous Time Fourier Series** or **CTFS**) Every periodic CT signal $x(t)$ with period T_0 (and hence frequency $f_0 = \frac{1}{T_0}$) that follows the Dirichlet conditions and has finite energy (discussion of which omitted) can be expressed as a converging infinite sum of fundamental periodic signals for some coefficients a_k.*

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{j2\pi f_0 kt} = \sum_{k=-\infty}^{\infty} a_k e^{j\Omega_0 kt}$$

*We call a_k the **Fourier Series Coefficients** of $x(t)$, and their values are given by the formula*

$$a_k = \frac{1}{T}\int_0^T x(t)e^{-j2\pi f_0 kt}dt = \frac{1}{T}\int_0^T x(t)e^{-jk\Omega_0 t}dt$$

In addition, Fourier came up with the more startling fact that every signal (even if they aren't periodic!) meeting a certain criteria can be expressed as an infinite *integral* of fundamental periodic signals! This is the concept of the Fourier Transform, which is formalized below.

Theorem 7. *(**Continuous Time Fourier Transform** or **CTFT**) Every CT signal $x(t)$ (not just periodic ones!) satisfying the condition that $\int_{-\infty}^{\infty} |x(t)|^2 dt < \infty$ (having finite energy) can be expressed as an infinite integral of fundamental periodic signals:*

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\Omega)e^{j\Omega t}d\Omega$$

*We call $X(j\Omega)$ the **Continuous Time Fourier Transform** of $x(t)$, which has an equation given by:*

$$X(j\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t}dt$$

This leads us to the idea of looking at signals from the frequency perspective. Fourier said that every signal, even though they aren't periodic themselves, is composed of *frequency components.* The Fourier Transform allows us to look at the strength of each of those frequency components and analyze/process signals accordingly, which is immensely useful. Finally, the CTFT has a nice property which we saw for LTI systems earlier.

Theorem 8. *The Continuous Time Fourier Transform is a linear operator. This means that if $x_1(t)$ has CTFT $X_1(j\Omega)$ and $x_2(t)$ has CTFT $X_2(j\Omega)$, then for all scalars λ, $x_1(t) + \lambda x_2(t)$ will have a CTFT $X_1(j\Omega) + \lambda X_2(j\Omega)$.*

6 The Frequency Domain Interpretation of DT Signals

In this section, we will go through a similar process of what we did last section for discrete time signals.

Definition 14. *The fundamental periodic DT signal over the field $F = \mathbb{C}$ is of the form*

$$x(t) = Ae^{j\omega n+\phi}$$

*Call ω the **frequency** of $x[n]$ in radians/sample. Call ϕ the **phase shift**.*

Exercise 2. *Compare the signals*

$$x_1[n] = cos(\pi n)$$
$$x_2[n] = cos(3\pi n)$$
$$x_3[n] = cos(5\pi n)$$

Is there anything interesting that happens?

This exercise shows an important result: frequencies in discrete time are fundamentally different from frequencies in continuous time. In discrete time, a frequency can only range from $-\pi$ to π, and afterwards, everything folds back into something inside this range. I will go over this in more detail during class. Though there is a digital analog to the Fourier Series, I will omit that in these notes and focus on the Fourier Transform.

Theorem 9. *(**Discrete Time Fourier Transform** or **DTFT**) Every DT signal $x[n]$ (not just periodic ones!) satisfying the condition that $\sum_{n=-\infty}^{\infty} |x[n]|^2 < \infty$ (having finite energy) can be expressed as an infinite integral of fundamental periodic signals:*

$$x[n] = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(e^{j\omega}) e^{j\omega n} d\omega$$

*We call $X(e^{j\omega})$ the **Discrete Time Fourier Transform** of $x[n]$, which has an equation given by:*

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

Note that a major difference between the DTFT and the CTFT is that the DTFT is periodic with period $N = 2\pi$. Therefore, it is usually convention to talk about the DTFT in the range $[-\pi, \pi]$.

Theorem 10. *The Discrete Time Fourier Transform is a linear operator. This means that if $x_1[n]$ has DTFT $X_1(e^{j\omega})$ and $x_2[n]$ has DTFT $X_2(e^{j\omega})$, then for all scalars λ, $x_1[n] + \lambda x_2[n]$ will have a CTFT $X_1(e^{j\omega}) + \lambda X_2(e^{j\omega})$.*

7 Introduction to Sampling: The Relationship Between CT and DT

So far, we have introduced the notion of looking at both continuous time and discrete time signals in terms of frequency instead of in terms of time. Intuitively, the CTFT is easier to understand than the DTFT, but in real life, signals that are stored in your computer are discrete time signals! It would be beneficial to have a way of understanding how frequencies in continuous time relate to frequencies in discrete time so we can process these signals using a digital device such as a computer.

Definition 15. *We can use **uniform pointwise sampling** to convert a CT signal $x(t)$ into a DT signal $x[n]$. In this sampling scheme, we take a value at every **sampling period** T and add it to our DT signal $x[n]$. Mathematically,*

$$x[n] = x(nT) = x(t)|_{t=nT}$$

*Call $f_s = \frac{1}{T}$ the **sampling frequency**.*

Sampling may seem like an operation that inherently loses information about the original CT signal. After all, it seems like to store all the information about a CT signal, we would need to know every single value at every single time. That would imply that the sampling frequency $f_s = \infty$, which is not feasible! However, it turns out that we can sample signals without losing any information, and the condition for doing so is much more realizable than we may initially think. The key is to recognize that we just have to make sure that we don't lose any information about the signal's Fourier Transform.

Definition 16. *We call a CT signal $x(t)$ with CTFT $X(j\Omega)$ **bandlimited** if there exists a frequency $\Omega_m = 2\pi f_m$ such that for all $\Omega > \Omega_m$, $X(j\Omega) = 0$. Call Ω_m the **bandwidth** of $x(t)$.*

Theorem 11. *(**The Shannon-Nyquist Sampling Theorem**) We can use uniform pointwise sampling to sample a CT signal $x(t)$ into DT signal $x[n]$ if and only if $x(t)$ is bandlimited. Let*

$$\Omega_m = 2\pi f_m$$

be the bandwidth of $x(t)$. A sufficient condition to sample $x(t)$ without losing information is to choose a sampling frequency

$$f_s \geq 2 f_m$$

*This condition is called the **Nyquist criterion**.*

Proof. Omitted. (Saved for your first DSP class if you're going to study Electrical Engineering in college!) □

Now that we know the sufficient condition for a signal to be sampled without losing any information, all we need is to be able to map CT frequencies into DT frequencies, and back again in order to really start doing digital signal processing! For that, the below theorem helps us out:

Theorem 12. *Suppose a CT signal $x(t)$ is sampled into a DT signal $x[n]$ using sampling frequency*

$$\Omega_s = 2\pi f_s = \frac{2\pi}{T}$$

Call $X(j\Omega)$ the CTFT of $x(t)$ and $X(e^{j\omega})$ the DTFT of $x[n]$. Suppose that the Nyquist criterion is satisfied. Then, for all frequencies Ω,

$$X(j\Omega) = \frac{1}{T} X(e^{j(\Omega T)})$$

In other words, the CT frequency Ω is related to the DT frequency ω by the relation

$$\omega = \Omega T$$

and there is a scaling of $\frac{1}{T}$ in going from CTFT to DTFT.

Proof. Omitted. □

8 Designing Systems to Process CT/DT Signals

Now that we looked into the frequency domain interpretation of signals in great detail, we would like to start processing signals using systems. In section three, we showed that the output of a Linear and Time Invariant System $y[n]$ given an input signal $x[n]$ is the convolution of x and the impulse response h,

$$y[n] = x[n] \star h[n]$$

Now that we know a little bit about the CTFT and the DTFT, we can use a powerful result to start designing H, or equivalently, $h[n]$ to start doing interesting things!

Theorem 13. *Convolution in the time domain is multiplication in the frequency domain. If*

$$x[n] \longleftrightarrow X(e^{j\omega})$$

and

$$h[n] \longleftrightarrow H(e^{j\omega})$$

then

$$x[n] \star h[n] \longleftrightarrow X(e^{j\omega})H(e^{j\omega})$$

(The same is true for CTFT)

This theorem states that taking a convolution with two signals is equivalent to first taking either the CTFT or the DTFT of the two signals, multiplying them together, and then finally taking an inverse CTFT or inverse DTFT to get back a resulting signal. This also means that designing a system to do something can be as simple as choosing a signal whose CTFT/DTFT does something you want and then taking the inverse of that to get your actual impulse response! Some interesting systems are called **filters**, and they can be designed to do different tasks

Definition 17. *A **low pass filter** is a filter designed to kill off frequencies that are higher than the frequency f_c, called the **cutoff frequency**.*

Definition 18. *A **band pass filter** is a filter designed to kill off frequencies that are between the frequencies f_{c1} and f_{c2}.*

Definition 19. *A **high pass filter** is a filter designed to kill off frequencies that are lower than the frequency f_c.*

Exercise 3. *Starting from the CTFT $H(j\Omega)$ and using the theorem above, try designing the impulse response of a high pass filter with a cutoff frequency of 500 Hz.*

9 Computational Aspects

With sampling and knowledge of how to design the CTFT/DTFT of systems from the past two sections, we can now (almost) design systems that process either DT or CT signals to run on a computer! One problem that we had is that the DTFT is a *continuous time* transform of a DT signal; the transform itself is a continuous time function! That won't help much if we want to do things on a computer; continuous time functions require infinite space on your computer, and that is not possible. We will approach a solution by sampling the DTFT.

Definition 20. *(The **Discrete Fourier Transform** or **DFT**) Suppose we want to sample the DTFT $X(e^{j\omega})$ of a DT signal $x[n]$. Choosing a value N and sampling at every $\frac{2\pi}{N}$ interval, we get the sequence*

$$X[k] = X(e^{j\omega})|_{\omega=\frac{2\pi}{N}k} = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}$$

*Call $X[k]$ from above the **N-point Discrete Fourier Transform**.*

With the DFT, we have the ability to do **spectral analysis** of signals by looking at the frequency domain of signals and doing processing on a computer. Clearly, as we choose larger and larger N, the more information about the DTFT we will be able to know from an N-point DFT of a signal $x[n]$. The trade-off is that as N increases, the longer it will take to compute the N-point DFT $X[k]$. For those into Computer Science, the time complexity of computing a DFT by the definition is $O(N^2)$, which is infeasible for large N. Luckily, certain **Fast Fourier Transform** algorithms exist that can compute an N-point DFT of a signal with time complexity $O(N\log_2(N))$ given that N is a power of two, which is much better than $O(N^2)$ and is the reason why signal processing using digital systems such as computers is acceptable for many tasks as opposed to specialized hardware which may be tedious to design.