ECE 310 Fall 2023

Lecture 3

Discrete-time systems

Corey Snyder

Learning Objectives

After this lecture, you should be able to:

- Define a discrete-time system.
- State the conditions for a systems to be linear, time-invariant, and causal.
- State the definition of a system being bounded-input bounded-output (BIBO) stable.
- Prove whether a given discrete-time system is linear, time-invariant, causal, or stable.

Recap from previous lecture

We covered important preliminaries in the previous lecture including complex numbers and common digital signal representations. This lecture will explain how to define discrete-time systems and how to characterize them as being linear/non-linear, time-invariant/time-varying, causal/non-causal, or stable/unstable.

1 Discrete-time systems

A *discrete-time system* is any computational process that maps a discrete-time input, $x[n]$, to a discrete-time output, $y[n]$. Let T define our abstract system or *operator* that maps $x[n]$ to $y[n]$. We can then notate our system as follows:

$$x[n] \overset{T}{\mapsto} y[n] \tag{1}$$

$$y[n] = T(x[n]). \tag{2}$$

In words, Eqn. 1 says that "$x[n]$ maps to $y[n]$ by operator T" while Eqn. 2 says "$y[n]$ is a function of $x[n]$ described by T". We will not frequently use this notation; however, it will be convenient when defining and proving the properties in Section 2 below.

Before continuing, let's define a couple example systems to show how flexible this notation is. One simple example is as follows:

$$y[n] = x[n] - x[n-1]. \tag{3}$$

This system simply computes the difference between the present input sample and the previous sample. In this example, T would be the two-sample difference on the right side of the equation. In practice, it could be useful for detecting large or fast changes in a signal. Now let's try a more abstract system:

$$y[n] = \text{HalfCenterCrop}\left(x[n]\right) \tag{4}$$

This system takes an input image $x[n]$ and crops down to the center of the image so that the output $y[n]$ is half as tall and half as wide. Here, our system T is the "HalfCenterCrop" operator and while this may look more like Python code than a discrete-time system, it still satisfies our definition for a digital system! This center cropping of an image is a well-defined mapping.

This brings us to an important point before we move on: **discrete-time systems are not just filters**. We will of course discuss filters plenty in this class; however, digital filters are just one example of discrete-time systems. Photoshopping tools, automatic speech recognition on your phone (after analog-to-digital conversion!), high-frequency trading algorithms, music recommendation systems, and (of course) de-noising filters are all discrete-time systems that fit within our definition!

2 Linearity, time-invariance, causality, and stability

There are several important properties we use to characterize discrete-time systems: specifically linearity, time-invariance, causality, and stability. We will define and provide examples for how to prove each of these properties in the following sections. In this course, we will frequently work with linear, time-invariant (LTI) systems as they offer nice properties for our signal processing framework in this class (as we will see in Lecture 4). It is important to note, however, that non-linear, time-varying, or non-causal systems appear throughout signal processing techniques and applications as well. We will point these examples out as we see them while others are outside the scope of this course or can be seen in our accompanying lab: ECE 311.

2.1 Linearity

A discrete-time system is *linear* if and only if it satisfies the following two conditions:

$$T(ax[n]) = aT(x[n]),\ \ a \in \mathbb{R} \tag{5}$$

$$T(x_1[n] + x_2[n]) = T(x_1[n]) + T(x_2[n]). \tag{6}$$

The condition given in Eqn. 5 is known as *homogeneity.* In plain words, homogeneity says that a scaled input, $ax[n]$, should produce an output that is the same as if we passed the original input, $x[n]$, to the system and scaled the output by a. Eqn. 6 states the condition of *additivity.* This says that adding two input signals, $x_1[n]$ and $x_2[n]$, then passing them to the system should give the same result as adding the respective outputs from applying the system to each input signal.

Instead of checking both conditions separately to prove linearity, we may also combine them into one condition known as *superposition*:

$$T(ax_1[n] + bx_2[n]) = aT(x_1[n]) + bT(x_2[n]),\ \ a, b \in \mathbb{R}. \tag{7}$$

Thus, a system T is linear if and only if it satisfies the condition of superposition. Let's try a couple quick examples now.

Exercise 1: Suppose we have the system given by $y[n] = x[n] - x[n-1]$ (this is our difference system from Eqn. 3). Is this system linear?.

We can check for linearity immediately by trying to prove superposition. Let $z[n] = ax_1[n] + bx_2[n]$:

$$\begin{aligned}
T(ax_1[n] + bx_2[n]) &\overset{?}{=} aT(x_1[n]) + bT(x_2[n]) \\
T(ax_1[n] + bx_2[n]) &= T(z[n]) \\
&= z[n] - z[n-1] \\
&= (ax_1[n] + bx_2[n]) - (ax_1[n-1] + bx_2[n-1]) \\
&= a(x_1[n] - x_1[n-1]) + b(x_2[n] - x_2[n-1]) \\
&= aT(x_1[n]) + bT(x_2[n]).\ \checkmark
\end{aligned}$$

Thus, the system is **linear**.

Exercise 2: Let our system T be given by $y[n] = (x[n])^p,\ p > 0$. Is this system linear?

We can try to prove linearity via superposition like we did in the previous exercise. However, let's start with just trying to prove homogeneity:

$$\begin{aligned} T(ax[n]) &\overset{?}{=} aT(x[n]) \\ T(ax[n]) &= (ax[n])^p \\ &= a^p(x[n])^p \\ &= a^pT(x[n]) \\ &\neq aT(x[n]) \text{ ✗} \end{aligned}$$

Thus, the system is **non-linear**.

2.2 Time-invariance

We say a discrete-time system is *time-invariant* or *shift-invariant* if:

$$y[n-n_0] = T(x[n-n_0]) \tag{8}$$

or

$$T(x)[n-n_0] = T(x[n-n_0]). \tag{9}$$

Time-invariance means that a shift by n_0 in our input yields the same shift in our output. The above two definitions both say this with slightly different depictions of the output. The left side of both definitions describe passing $x[n]$ to a discrete-time system *then* shifting by n_0. The right side instead shows that we shift the input signal by n_0 *then* pass this shifted input to the system. The definition of time-invariance may seem simple and intuitive, but often time-invariance is the hardest to prove of the properties in this lecture. Let's look at one tricky example:

Exercise 3: Suppose our system T is described by $y[n] = x[|n|]$. Is this system shift-invariant?

$$\begin{aligned} y[n-n_0] &\overset{?}{=} T(x[n-n_0]) \\ y[n-n_0] &= x[|n-n_0|] \\ T(x[n-n_0]) &= x[|n|-n_0|]. \end{aligned}$$

We see that the two sides of the relation are not equivalent, thus the system is **time-varying**.

Exercise 3 provides an important example of how we treat systems where T acts on the indices of the input signal (or at least one term). More concretely, let $f(n)$ denote some function on the time variable of $x[n]$ such that $T(x[n]) = x\left[f(n)\right]$. Time-invariance then checks if

$$y[n-n_0] \equiv x[f(n-n_0)] \overset{?}{=} x[f(n)-n_0] \equiv T(x[n-n_0]). \tag{10}$$

Remember that the "action" of our system is to apply some function to the time variable of the input signal, i.e. n. The left side has us compute $y[n-n_0]$. Thus, the left side evaluates if we act on the signal *then* shift the output. Therefore, we compute $y[n] = x[f(n)]$ then replace every n with $n - n_0$. Meanwhile, the right side evaluates if we shift the input *then* act on the time variable. It's important to note that "acting on the time variable" refers to n since it is actually our time variable while n_0 is just a constant. Exercise 3 provides an example of this nuance playing out with $f(n) = |n|$. Let's look at a couple more examples to make this clear.

Exercise 4: Is $y[n] = x[n] - x[n-1]$ time-invariant?

$$\begin{aligned} y[n-n_0] &\overset{?}{=} T(x[n-n_0]) \\ y[n-n_0] &= x[(n-n_0)] - x[(n-n_0)-1] \\ &= x[n-n_0] - x[n-n_0-1] \\ T(x[n-n_0]) &= x[n-n_0] - x[n-1-n_0] \\ &= x[n-n_0] - x[n-n_0-1]. \end{aligned}$$

The two results are equivalent here, thus the system is **time-invariant**. Notice how we group the indices within each $x[n]$ term differently for the output and input sides of the relation to make clear whether the shift occurs before or after the system T acts on the signal.

Exercise 5: Is $y[n] = nx[3n]$ time-invariant?

$$\begin{aligned} y[n-n_0] &\overset{?}{=} T(x[n-n_0]) \\ y[n-n_0] &= (n-n_0)x[3(n-n_0)] \\ T(x[n-n_0]) &= nx[3n-n_0]. \end{aligned}$$

These two results are not equal, thus the system is **time-varying**. Note how we only replace the n in front of the $x[n]$ with $n - n_0$ for the shifted output. The shifted input only adds a shift by n_0 inside of the actual input signal's argument.

2.3 Causality

A discrete-time system is *causal* if the system output only depends on present or previous input and output samples. To put it another way, our system is causal if the output does not depend on future samples.

Exercise 6: Let our system be given by $y[n] = x[n] + x[n-2] - x[n-4]$. Is this system causal?

We can check for causality by comparing the output index to the indices of each term in our system:

$$\begin{aligned} &\text{Output } \ n \overset{?}{\geq} \text{System term } n \\ &\text{Term 1: } n \geq n \\ &\text{Term 2: } n \geq n-2 \\ &\text{Term 3: } n \geq n-4. \ \checkmark \end{aligned}$$

Thus, our system is **causal**.

The above example gives a straightforward case of proving a system is causal. When showing a system is non-causal, we often can simply show that for some value of n the output $y[n]$ depends on a future input sample. For example, consider $n = -3$ for $y[n] = x[|n|]$.

2.4 Stability

Lastly, we will provide the definition of stability in discrete-time systems. We will revisit stability in future lectures from other perspectives. For now, a digital system T is *bounded-input bounded-output* (BIBO) stable if

$$\text{For any } x[n] \text{ such that } |x[n]| < \beta,\ \forall n,\ |T(x[n])| < \alpha,\ \forall\ n,\ \text{where } 0 \leq \alpha < \infty, 0 \leq \beta < \infty. \tag{11}$$

The statement in Eqn. 11 says a lot, but it is an intuitive definition. A system is BIBO stable if for any bounded input signal, i.e. a signal that takes on finite values for all n, the system output is also bounded (magnitude less than some finite value α) for all indices n. Above, the symbol $\forall$ denotes "for all".

Exercise 7: Suppose we are given the system $y[n] = x^{10}[n] + e^{x[n]}$. Is this system BIBO stable?

By the definition of BIBO stability, we may say that our input $|x[n]| < \beta < \infty$ for all n. This means that

$$\begin{aligned} |y[n]| &= |x^{10}[n] + e^{x[n]}| \\ &\leq |x[n]|^{10} + e^{|x[n]|} \\ &< \beta^{10} + e^{\beta} \\ &< \infty, \ \forall n. \ \checkmark \end{aligned}$$

Thus, our system is **BIBO stable** where $|y[n]|$ is guaranteed to be less than $\alpha = \beta^{10} + e^{\beta}$.