Lecture 12: Linearity and Shift-Invariance

Mark Hasegawa-Johnson

ECE 401: Signal and Image Analysis, Fall 2021

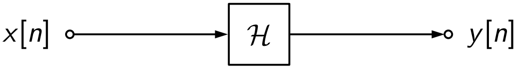
Outline

What is a System?

A **system** is anything that takes one signal as input, and generates another signal as output. We can write

$$x[n] \xrightarrow{\mathcal{H}} y[n]$$

which means



Example: Averager

For example, a weighted local averager is a system. Let's call it system $\mathcal{A}$.

$$x[n] \stackrel{\mathcal{A}}{\longrightarrow} y[n] = \sum_{m=0}^{6} g[m]x[n-m]$$

Example: Time-Shift

A time-shift is a system. Let's call it system $\mathcal{T}$.

$$x[n] \stackrel{\mathcal{T}}{\longrightarrow} y[n] = x[n-1]$$

Example: Square

If you calculate the square of a signal, that's also a system. Let's call it system $\mathcal{S}$:

$$x[n] \stackrel{\mathcal{S}}{\longrightarrow} y[n] = x^2[n]$$

Example: Add a Constant

If you add a constant to a signal, that's also a system. Let's call it system $\mathcal{C}$:

$$x[n] \stackrel{\mathcal{C}}{\longrightarrow} y[n] = x[n] + 1$$

Example: Window

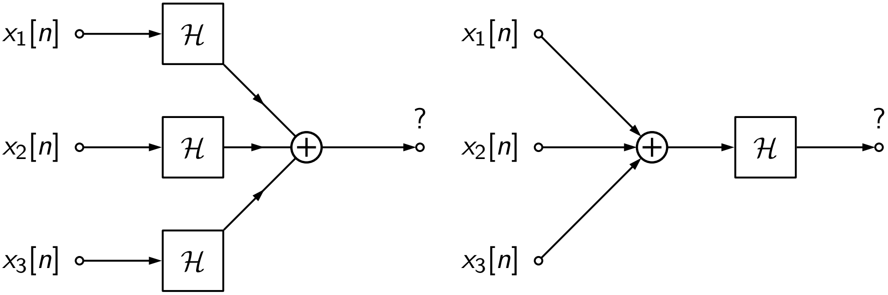
If you chop off all elements of a signal that are before time 0 or after time $N-1$ (for example, because you want to put it into an image), that is a system:

$$x[n] \stackrel{\mathcal{W}}{\longrightarrow} y[n] = \begin{cases} x[n] & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases}$$

Outline

Linearity

A system is **linear** if these two algorithms compute the same thing:



Linearity

A system $\mathcal{H}$ is said to be **linear** if and only if, for any $x_1[n]$ and $x_2[n]$,

$$x_1[n] \xrightarrow{\mathcal{H}} y_1[n]$$
$$x_2[n] \xrightarrow{\mathcal{H}} y_2[n]$$

implies that

$$x[n] = x_1[n] + x_2[n] \xrightarrow{\mathcal{H}} y[n] = y_1[n] + y_2[n]$$

In words: a system is **linear** if and only if, for every pair of inputs $x_1[n]$ and $x_2[n]$, (1) adding the inputs and then passing them through the system gives exactly the same effect as (2) passing both inputs through the system, and **then** adding them.

Special case of linearity: Scaling

Notice, a special case of linearity is the case when $x_1[n] = x_2[n]$:

$$x_1[n] \stackrel{\mathcal{H}}{\longrightarrow} y_1[n]$$
$$x_1[n] \stackrel{\mathcal{H}}{\longrightarrow} y_1[n]$$

implies that

$$x[n] = 2x_1[n] \stackrel{\mathcal{H}}{\longrightarrow} y[n] = 2y_1[n]$$

So if a system is linear, then **scaling the input** also **scales the output**.

Example: Averager

Let's try it with the weighted averager.

$$x_1[n] \xrightarrow{\mathcal{A}} y_1[n] = \sum_{m=0}^{6} g[m]x_1[n-m]$$

$$x_2[n] \xrightarrow{\mathcal{A}} y_2[n] = \sum_{m=0}^{6} g[m]x_2[n-m]$$

Then:

$$\begin{aligned} x[n] = x_1[n] + x_2[n] &= \sum_{m=0}^{6} g[m]\left(x_1[n-m] + x_2[n-m]\right) \\ &= \left(\sum_{m=0}^{6} g[m]x_1[n-m]\right) + \left(\sum_{m=0}^{6} g[m]x_2[n-m]\right) \\ &= y_1[n] + y_2[n] \end{aligned}$$

. . . so a weighted averager is a **linear system**.

Example: Square

A squarer is just obviously nonlinear, right? Let's see if that's true:

$$x_1[n] \xrightarrow{\mathcal{S}} y_1[n] = x_1^2[n]$$
$$x_2[n] \xrightarrow{\mathcal{S}} y_2[n] = x_2^2[n]$$

Then:

$$\begin{aligned} x[n] = x_1[n] + x_2[n] \xrightarrow{\mathcal{A}} y[n] &= x^2[n] \\ &= (x_1[n] + x_2[n])^2 \\ &= x_1^2[n] + 2x_1[n]x_2[n] + x_2^2[n] \\ &\neq y_1[n] + y_2[n] \end{aligned}$$

...so a squarer is a **nonlinear system**.

Example: Add a Constant

This one is tricky. Adding a constant seems like it ought to be linear, but it's actually **nonlinear**. Adding a constant is what's called an **affine** system, which is not necessarily linear.

$$x_1[n] \xrightarrow{\mathcal{C}} y_1[n] = x_1[n] + 1$$
$$x_2[n] \xrightarrow{\mathcal{C}} y_2[n] = x_2[n] + 1$$

Then:

$$\begin{aligned} x[n] = x_1[n] + x_2[n] \xrightarrow{\mathcal{A}} y[n] &= x[n] + 1 \\ &= x_1[n] + x_2[n] + 1 \\ &\neq y_1[n] + y_2[n] \end{aligned}$$

. . . so adding a constant is a **nonlinear system**.

What about the real world?

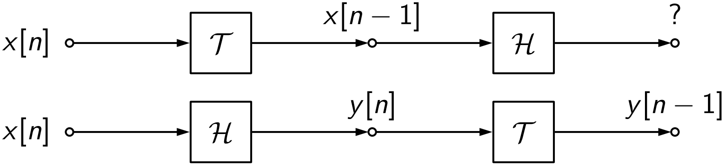
Suppose you're showing people images $x[n]$, and measuring their brain activity $y[n]$ as a result. How can you tell if this system is linear?

- Show them one image, call it $x_1[n]$. Measure the resulting brain activity, $y_1[n]$.
- Show them another image, $x_2[n]$. Measure the brain activity, $y_2[n]$.
- Show them $x[n] = x_1[n] + x_2[n]$. Measure $y[n]$. Is it equal to $y_1[n] + y_2[n]$?
- Repeat this experiment with lots of different images, and their sums, until you are convinced that the system is linear (or not).

Outline

Shift Invariance

A system $\mathcal{H}$ is **shift-invariant** if these two algorithms compute the same thing (here $\mathcal{T}$ means "time shift"):



Shift Invariance

A system $\mathcal{H}$ is said to be **shift-invariant** if and only if, for every $x_1[n]$,

$$x_1[n] \stackrel{\mathcal{H}}{\longrightarrow} y_1[n]$$

implies that

$$x[n] = x_1[n - n_0] \stackrel{\mathcal{H}}{\longrightarrow} y[n] = y_1[n - n_0]$$

In words: a system is **shift-invariant** if and only if, for any input $x_1[n]$, (1) shifting the input by some number of samples n_0, and then passing it through the system, gives exactly the same result as (2) passing it through the system, and then shifting it.

Example: Averager

Let's try it with the weighted averager.

$$x_1[n] \xrightarrow{\mathcal{A}} y_1[n] = \sum_{m=0}^{6} g[m]x_1[n-m]$$

Then:

$$\begin{aligned} x[n] = x_1[n-n_0] \xrightarrow{\mathcal{A}} y[n] &= \sum_{m=0}^{6} g[m]x[n-m] \\ &= \sum_{m=0}^{6} g[m]x_1\left[(n-m)-n_0\right] \\ &= \sum_{m=0}^{6} g[m]x_1\left[(n-n_0)-m\right] \\ &= y_1[n-n_0] \end{aligned}$$

...so a weighted averager is a **shift-invariant system**.

Example: Square

Squaring the input is a nonlinear operation, but is it shift-invariant? Let's find out:

$$x_1[n] \xrightarrow{\mathcal{S}} y_1[n] = x_1^2[n]$$

Then:

$$\begin{aligned} x[n] = x_1[n-n_0] \xrightarrow{\mathcal{A}} y[n] &= x^2[n] \\ &= (x_1[n-n_0])^2 \\ &= x_1^2[n-n_0] \\ &= y_1[n-n_0] \end{aligned}$$

...so computing the square is a **shift-invariant system**.

Example: Windowing

How about windowing, e.g., in order to create an image?

$$x_1[n] \xrightarrow{\mathcal{W}} y_1[n] = \begin{cases} x_1[n] & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases}$$

If we shift the **output**, we get

$$y_1[n-n_0] = \begin{cases} x_1[n-n_0] & n_0 \le n \le N-1+n_0 \\ 0 & \text{otherwise} \end{cases}$$

…but if we shift the **input** ($x[n] = x_1[n-n_0]$), we get

$$y[n] = \begin{cases} x[n] & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases} = \begin{cases} x_1[n-n_0] & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases}$$
$$\ne y_1[n-n_0]$$

…so windowing is a **shift-varying system** (not shift-invariant).

How about the real world?

Suppose you're showing images $x[n]$, and measuring the neural response $y[n]$. How do you determine if this system is shift-invariant?

- Show an image $x_1[n]$, and measure the neural response $y_1[n]$.
- Shift the image by n_0 columns to the right, to get the image $x[n] = x_1[n - n_0]$. Show people $x[n]$.
- Is the resulting neural response exactly the same, but shifted to a slightly different set of neurons (shifted "to the right?") If so, then the system may be shift-invariant!
- Keep doing that, with many different images and many different shifts, until you're convinced the system is shift-invariant.

Outline

LSI Systems and Convolution

We care about linearity and shift-invariance because of the following remarkable result:

LSI Systems and Convolution

Let $\mathcal{H}$ be any system,

$$x[n] \stackrel{H}{\longrightarrow} y[n]$$

If $\mathcal{H}$ is linear and shift-invariant, then whatever processes it performs can be equivalently replaced by a convolution:

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

Impulse Response

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

The weights $h[m]$ are called the "impulse response" of the system. We can measure them, in the real world, by putting the following signal into the system:

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

and measuring the response:

$$\delta[n] \xrightarrow{H} h[n]$$

Convolution: Proof

1. $h[n]$ is the impulse response.

$$\delta[n] \xrightarrow{H} h[n]$$

2. The system is **shift-invariant**, therefore

$$\delta[n-m] \xrightarrow{H} h[n-m]$$

3. The system is **linear**, therefore **scaling the input by a constant** results in **scaling the output by the same constant**:

$$x[m]\delta[n-m] \xrightarrow{H} x[m]h[n-m]$$

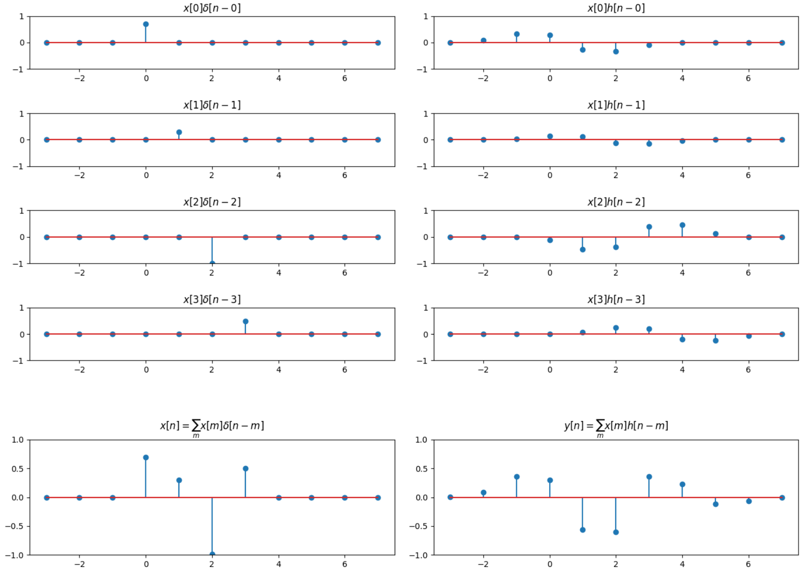
4. The system is **linear**, therefore **adding input signals** results in **adding the output signals**:

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

Convolution: Proof (in Words)

- The input signal, $x[n]$, is just a bunch of samples.
- Each one of those samples is a scaled impulse, so each one of them produces a scaled impulse response at the output.
- Convolution = add together those scaled impulse responses.

Convolution: Proof (in Pictures)



Outline

Written Example

Prove that differentiation, $y(t) = \frac{dx}{dt}$, is a linear shift-invariant system (in terms of t as the time index, instead of n).

Outline

Summary

- A system is **linear** if and only if, for any two inputs $x_1[n]$ and $x_2[n]$ that produce outputs $y_1[n]$ and $y_2[n]$,

$$x[n] = x_1[n] + x_2[n] \xrightarrow{\mathcal{H}} y[n] = y_1[n] + y_2[n]$$

- A system is **shift-invariant** if and only if, for any input $x_1[n]$ that produces output $y_1[n]$,

$$x[n] = x_1[n - n_0] \xrightarrow{\mathcal{H}} y[n] = y_1[n - n_0]$$

- If a system is **linear and shift-invariant** (LSI), then it can be implemented using convolution:

$$y[n] = h[n] * x[n]$$

  where $h[n]$ is the impulse response:

$$\delta[n] \xrightarrow{\mathcal{H}} h[n]$$