# *EE120 - Fall'19 - Lecture 1 Notes*[1]

*Murat Arcak*

*29 August 2019*

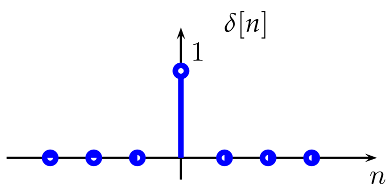


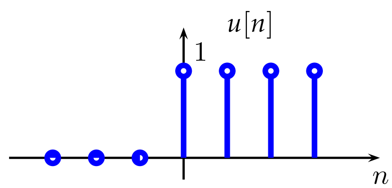
## *Signals*

Signals are functions of one or more variables, *e.g.* time in speech signals, two spatial variables in images, or both time and spatial variables in videos. In this course we will most often deal with functions of a single variable and generally refer to this variable as time.

A continuous-time signal, denoted $x(t)$, depends on a real-valued time variable $t$, and a discrete-time signal, $x[n]$, depends on the integer-valued variable $n$ that indexes instants of time. Below are two special discrete-time signals we will commonly use:

$$\text{Unit Impulse} : \delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \qquad \text{Unit Step} : u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

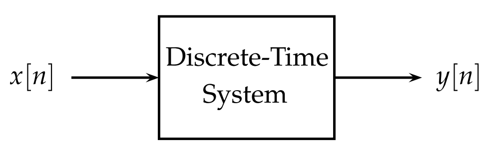


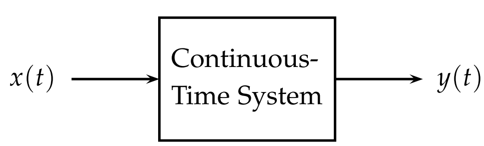




Note that $\delta[n] = u[n] - u[n-1]$, and $u[n] = \sum_{k=0}^{\infty} \delta[n-k]$.

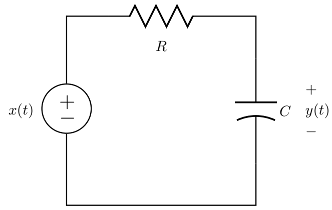
## *Systems*

For the purposes of this course, a system is defined as a process by which input signals are transformed to output signals. Inputs are typically denoted as $x$ and outputs as $y$.



Example: An RC circuit with capacitor voltage as the output:




$$C\frac{dy(t)}{dt} = \frac{-y(t) + x(t)}{R}$$

Example: A moving average filter:

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

*Memory*

A system is called memoryless if its output at a given time depends on the input only at that time.

Example: A resistor (with current and voltage interpreted as input and output) is memoryless, while the RC circuit above has memory due to the capacitor. Likewise, the moving average filter above has memory, as the output depends on $x[n-1]$ and $x[n+1]$, and would require memory registers to implement.

*Causality*

A system is called causal if its output depends on the input at present and past times only, not on future times.

Example: The RC circuit above is causal. The moving average filter is not causal, as $y[n]$ depends on the future input $x[n+1]$.

*Stability*

A system is called stable if all bounded inputs generate bounded outputs. It is unstable if not stable; that is, if there exists a bounded input for which the output grows unbounded.

Example: The moving average filter above is stable, since the output is simply an average of input values and remains bounded when the input is bounded.

Example: The "accumulator" system, defined by

$$y[n] - y[n-1] = x[n],\ y[-1] = 0 \tag{2}$$

when $x[n] = 0$ for $n < 0$ has the solution

$$y[n] = x[0] + x[1] + \cdots + x[n],\ n \geq 0, \tag{3}$$

which can grow unbounded with bounded inputs, such as the unit step input which gives $y[n] = n + 1$. The continuous-time analogue of the accumulator is the integrator:

$$\frac{dy(t)}{dt} = x(t), \; y(0) = 0,$$

which is likewise unstable.

### *Linearity*

A system is called linear if it satisfies these two conditions:
1. Scaling: For any input-output pair $x(t) \to y(t)$ and constant $a$,

$$ax(t) \to ay(t). \tag{4}$$

2. Superposition: For any two input-output pairs $x_1(t) \to y_1(t)$, $x_2(t) \to y_2(t)$,

$$x_1(t) + x_2(t) \to y_1(t) + y_2(t). \tag{5}$$

Corollary: If the input to a linear system is 0, the output must be 0.
*Proof.* Choose $a = 0$ in the scaling property.

Example: The moving average filter above is linear.

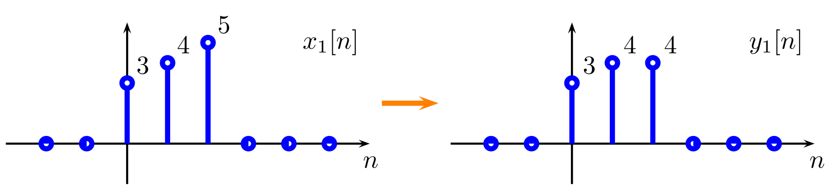
It is clear from (1) that if we scale the input, the output gets scaled by the same factor. The superposition property also holds because, if we apply the input $x[n] = x_1[n] + x_2[n]$, then from (1), $y[n]$ is given by

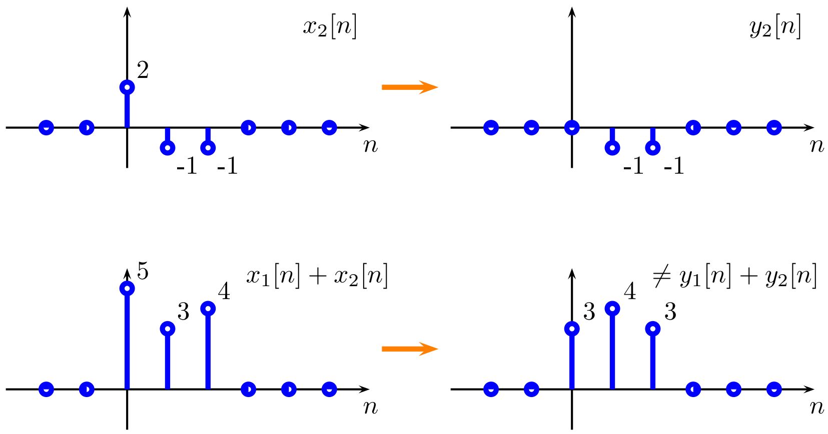
$$\begin{aligned}
&\frac{1}{3}\left((x_1[n-1] + x_2[n-1]) + (x_1[n] + x_2[n]) + (x_1[n+1] + x_2[n+1])\right) \\
&= \frac{1}{3}\left(x_1[n-1] + x_1[n] + x_1[n+1]\right) + \frac{1}{3}\left(x_2[n-1] + x_2[n] + x_2[n+1]\right) \\
&= y_1[n] + y_2[n].
\end{aligned}$$

Example: By contrast, the median filter:

$$y[n] = med\{x[n-1], x[n], x[n+1]\} \tag{6}$$

is nonlinear. Here are two input-output pairs that fail the superposition property:






*Time-Invariance:*

A system is called time-invariant if a time shift in the input results is an identical time shift in the output:

$$x(t-T) \rightarrow y(t-T)$$

for any input-output pair and any amount of shift $T$.

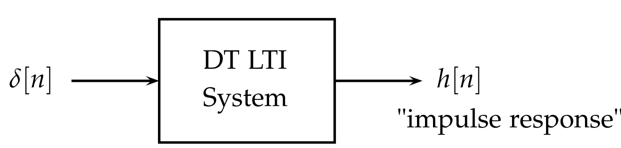
The moving average and median filters above are both time-invariant because the rule for generating the output (take the average or median of previous, current, and next input) does not change with time. On the other hand, the system $y[n] = nx[n]$ is time-varying, as it applies a different rule at each time (multiply the input with time $n$).

## *Linear Time-Invariant (LTI) Systems and Convolution*

Section 2.1 *in* Oppenheim & Willsky

Systems that are both linear and time-invariant, referred to as LTI systems, are amenable to powerful analysis tools that we will study in this course. In particular they have the remarkable property that knowing the response to a unit impulse input is enough to predict the response to any other input.

To see this in discrete-time, let $h[n]$ denote the response of a LTI system to the unit impulse $\delta[n]$:



and the rewrite the input $x[n]$ as

$$x[n] = ... + x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + ... \qquad (7)$$

Since $\delta[n] \to h[n]$, by time-invariance: $\delta[n-k] \to h[n-k]$ for any $k$. Then, by linearity:

$$y[n] = ... + x[-1]h[n+1] + x[0]h[n] + x[1]h[n-1] + ... \qquad (8)$$

Thus, for input $x[n]$, the output is:

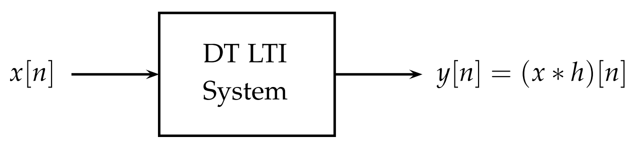
$$\boxed{y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]} \qquad (9)$$

which is a compact way of writing (8).

The operation on the right-hand side of (10) is called the "convolution" of signals $x$ and $h$, denoted $x * h$:

$$\boxed{(x*h)[n] := \sum_{k=-\infty}^{\infty} x[k]h[n-k].} \qquad (10)$$

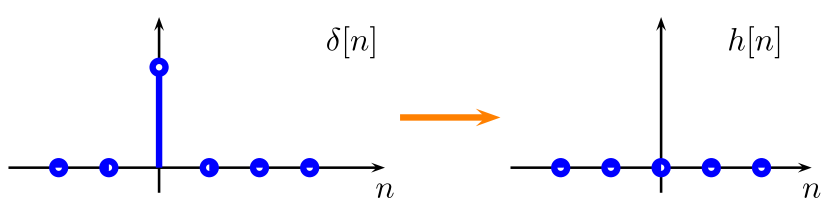
The following figure summarizes our conclusion:




In the next lecture we will see an analogous result for continuous-time systems.

Example: For the accumulator (2) the impulse response is the unit step, $u[n]$. Convince yourself that the convolution of $h[n] = u[n]$ and an input $x[n]$ (where $x[n] = 0$ for $n < 0$) recovers the solution (3).

Example: For the median filter (6),




Since the system is nonlinear, we can't use convolution to predict the output. Indeed, since $h[n] = 0$ for all $n$, its convolution with any input $x$ is $(x*h)[n] = 0$ for all $n$, which is clearly not the output of the median filter for all inputs.