

Linear Systems and Signals

Convolution properties

Anand D. Sarwate

Department of Electrical and Computer Engineering
Rutgers, The State University of New Jersey

2020



Learning objectives

The learning objective for this section is:

- ~~apply~~ properties of the convolution formula
understand



Many cheerful facts about the convolution

We write $y[n] = (\underline{x} * \underline{h})[n]$ or $\underline{y} = \underline{x} * \underline{h}$ for

$$y[n] = \sum_{k=-\infty}^{\infty} \underline{x[k]h[n-k]} = (\underline{x * h})[n]$$

The convolution has some nice properties:

- *Commutative:* $(\underline{x * h}) = (\underline{h * x})$

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

- *Associative:* $(\underline{x * h}) * \underline{g} = \underline{x * (h * g)}$.
- *Distributive:* $\underline{x * (g + h)} = (\underline{x * g}) + (\underline{x * h})$.



Commutativity: $(x * h) = (h * x)$

- ① Start with the definition:

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

Handwritten notes: A blue arrow points from $n-k$ to m with the label $= m$. The summation index $k=-\infty$ is underlined in blue.

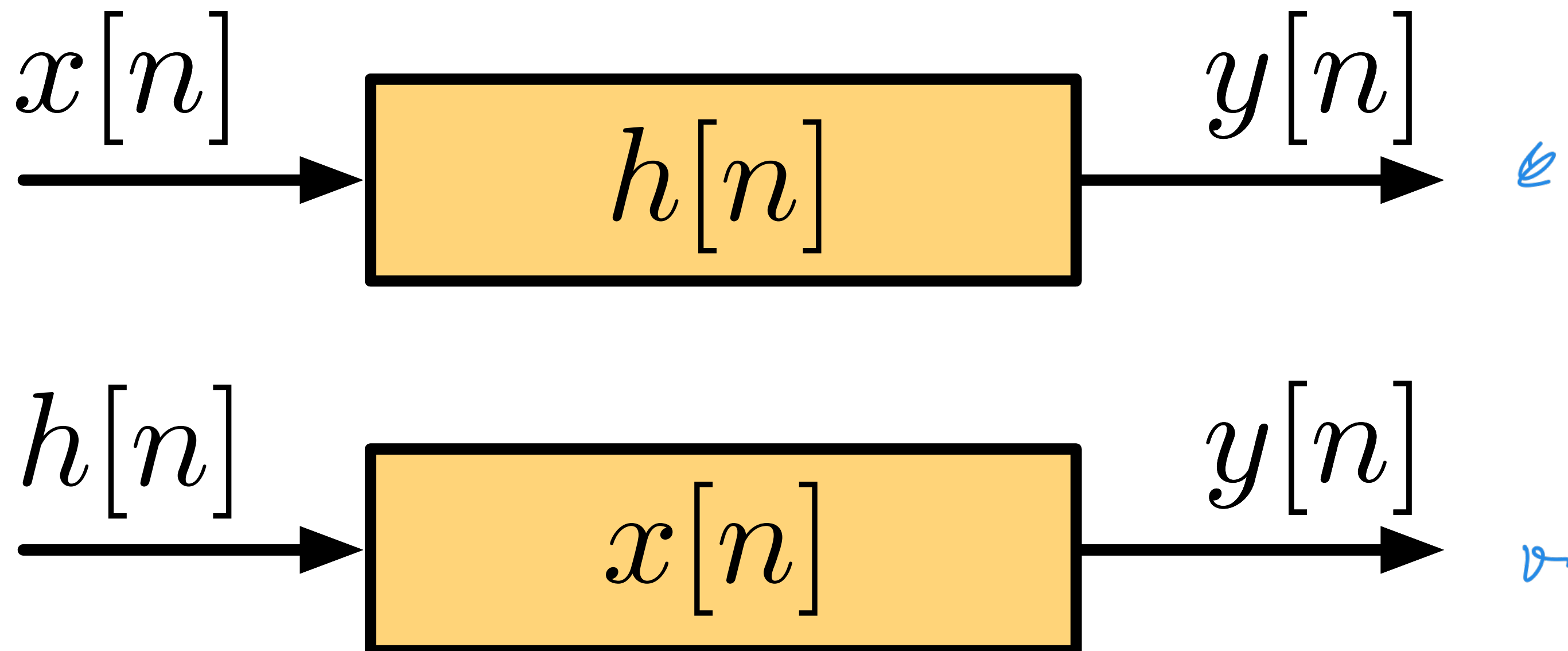
- ② Set $m = n - k$ so $k = n - m$. For any fixed n , as $k = -\infty$ to $+\infty$, m goes from $+\infty$ to $-\infty$:

$$\begin{aligned} (h * x)[n] &= \sum_{m=-\infty}^{\infty} h[n-m]x[m] \\ &= \sum_{m=-\infty}^{\infty} x[m]h[n-m] \\ &= (x * h)[n] \end{aligned}$$

Handwritten notes: In the first equation, $n-m$ and m are underlined in blue. In the second equation, $h[n-m]$ is underlined in blue. In the third equation, the final result is underlined in blue.

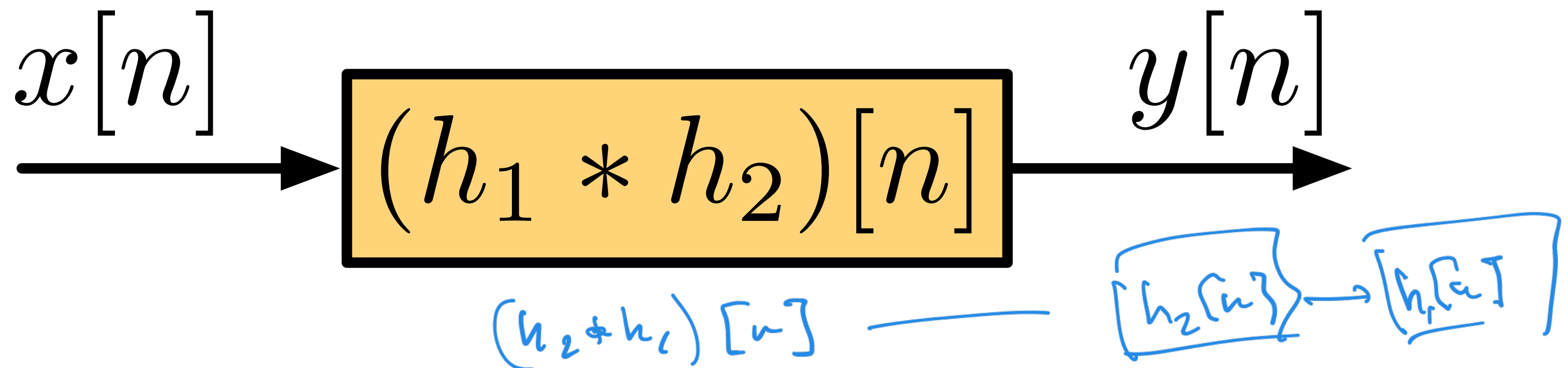
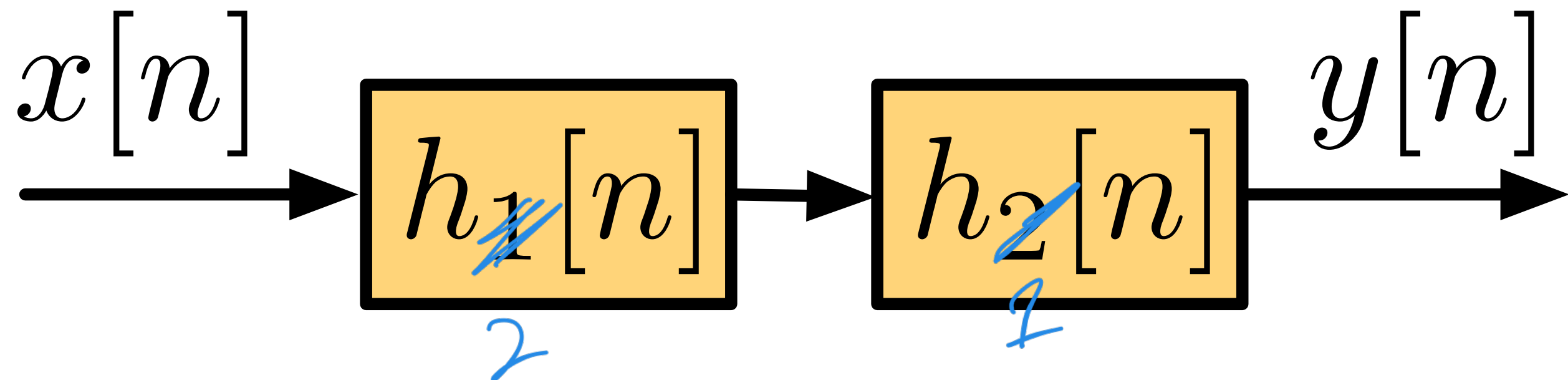


Why is commutativity useful?



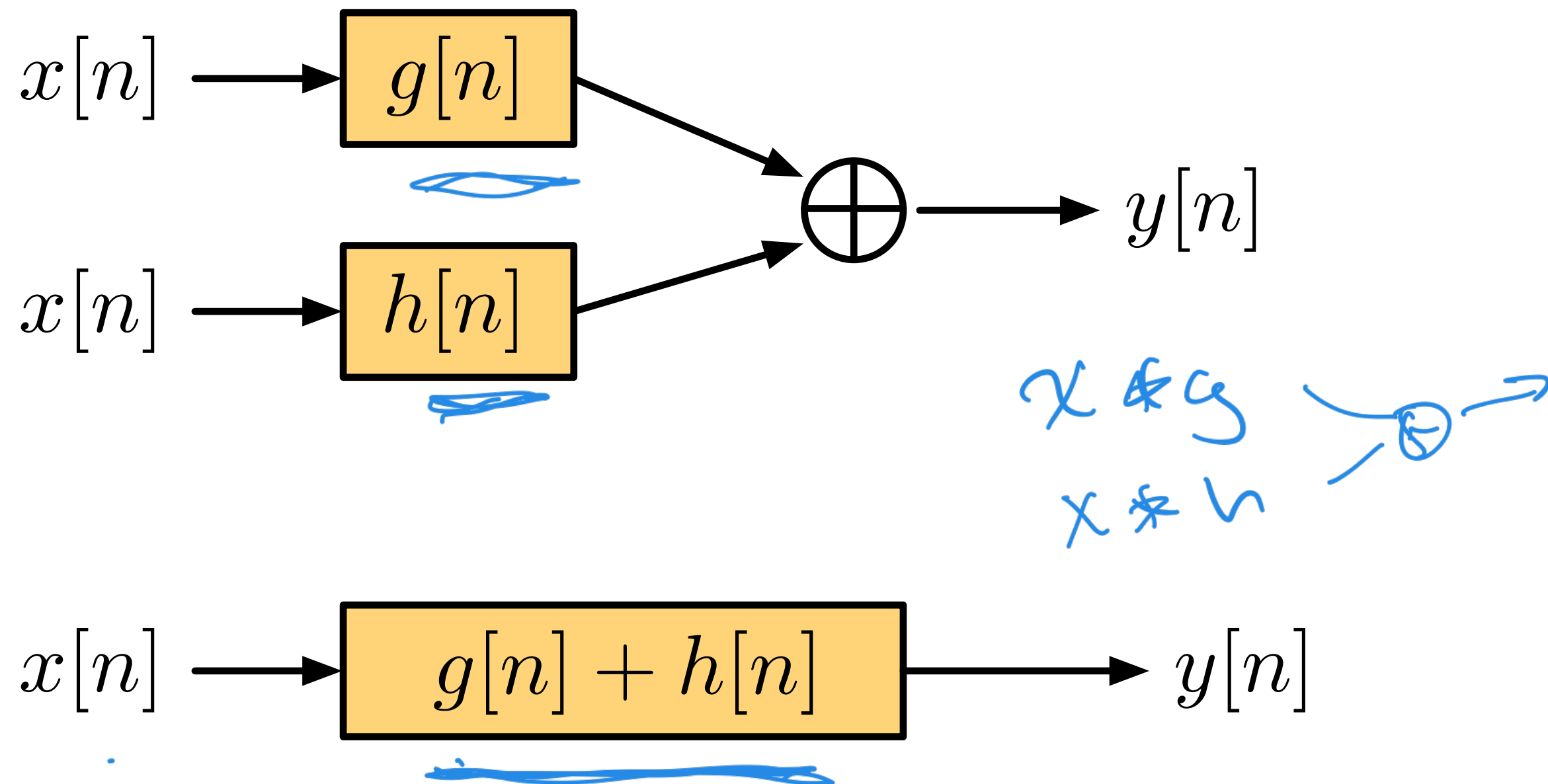
- Sometimes the calculation is easier one way than another.
- Can also think of $h[n]$ as a signal: lets us manipulate block diagrams.

Why is associativity useful?



- Merge/split cascades of systems
- With commutativity, swap the order of systems

Why is distributivity useful?



- Understand what systems are doing by looking at what each part does separately.
- Simplify block diagrams to get overall input/output behavior

Some language around DT LTI systems

- The **impulse response** is the output of the system when the input is $\delta[n]$, the unit impulse function.
- We often call an LTI system or its impulse response a **filter**.
- If $h[n]$ is only $\neq 0$ for a finite number of time points, we call it a **finite impulse response (FIR)** filter.
- If $h[n]$ is $\neq 0$ for an infinite number of time points, we call it an **infinite impulse response (IIR)** filter.

Handwritten sketch of a finite impulse response (FIR) filter, showing a sequence of discrete impulses (circles) that are non-zero for a finite duration, followed by the label FIR.

$$h[n] = \alpha^n u[n]$$

Handwritten label IIR below the equation.



The impulse response is everything

Critical fact: The output of a DT LTI system is only a function of the **input signal** and **impulse response**.

- We can find all of the system properties from the impulse response: causality, stability, etc.
- By relating the values in the impulse response relates to the what the system *does*, we can start to design systems for specific tasks.

