2.2 Arbitrary Inputs — Convolution Summation

read H & vV
2.1
2.2 up to conv. int. p. 84

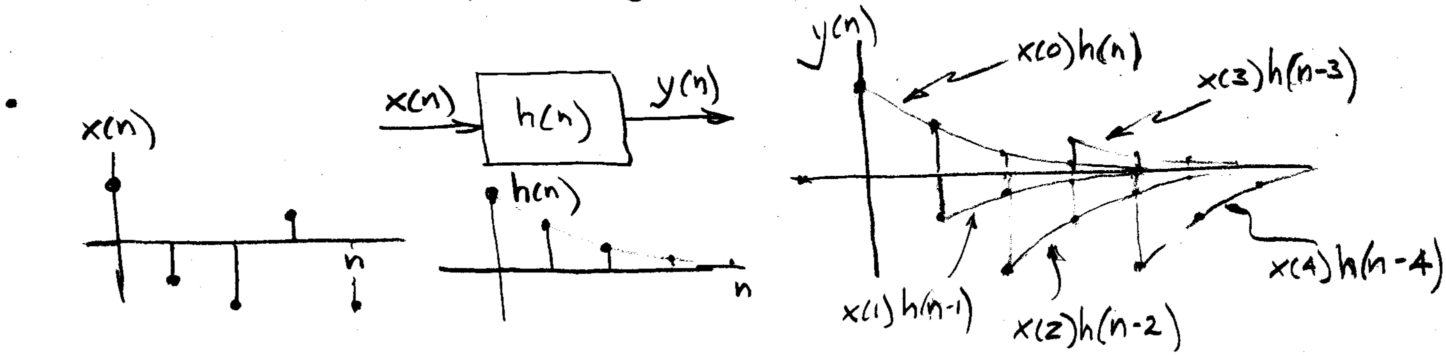
- In the previous section, we expressed an arbitrary input as a sum of weighted, delayed unit pulses.

$$x(n) = x(0)\,\delta(n) + x(1)\,\delta(n-1) + x(2)\,\delta(n-2) + \cdots$$

and found that the output could be written

$$y(n) = x(0)\,h(n) + x(1)\,h(n-1) + x(2)\,h(n-2) + \cdots$$

In this section, we'll go into more detail.



At any time n, we have

$$y(n) = x(0)h(n) + x(1)h(n-1) + \cdots + x(n-1)h(1) + x(n)h(0)$$

or $$y(n) = \sum_{i=0}^{n} x(i)\,h(n-i) \quad \text{or} \quad y(n) = \sum_{k=0}^{n} x(n-k)\,h(k)$$

These summation limits ensure both sequences are treated as causal — but if we define $x(n)$ and $h(n)$ to be zero for $n<0$, then

$$y(n) = \sum_{i=-\infty}^{\infty} x(i)\,h(n-i) \quad \text{or just} \quad \sum_{i} x(i)\,h(n-i)$$

- This relation is the convolution summation

$$y(n) = \sum_i x(i)\,h(n-i) = \sum_k x(n-k)\,h(k)$$

and it's denoted

$$y(n) = x(n) \circledast h(n)$$

Convolution is the basic time domain I/O relationship for LTI systems.

- Example $h(n) = 1, 2, 0, 0, 0, \ldots$

With input $x(n) = 1, -1, 2, 3, 0, 0, \ldots$ what is the output?

Do it by convolution $y(n) = \sum_{i=0}^{n} x(i)\,h(n-i)$

$$y(0) = x(0)\,h(0) \qquad = 1$$
$$y(1) = x(0)\,h(1) + x(1)\,h(0) \qquad = 1$$
$$y(2) = \cancel{x(0)\,h(2)} + x(1)\,h(1) + x(2)\,h(0) \qquad = 0$$
$$y(3) = \cancel{x(0)\,h(3)} + \cancel{x(1)\,h(2)} + x(2)\,h(1) + x(3)\,h(0) \qquad = 7$$

Slide $\boxed{2 \quad 1}$ $= h(n-i)$

across $\boxed{1 \quad -1 \quad 2 \quad 3 \quad 0 \quad 0}$ $= x(n)$

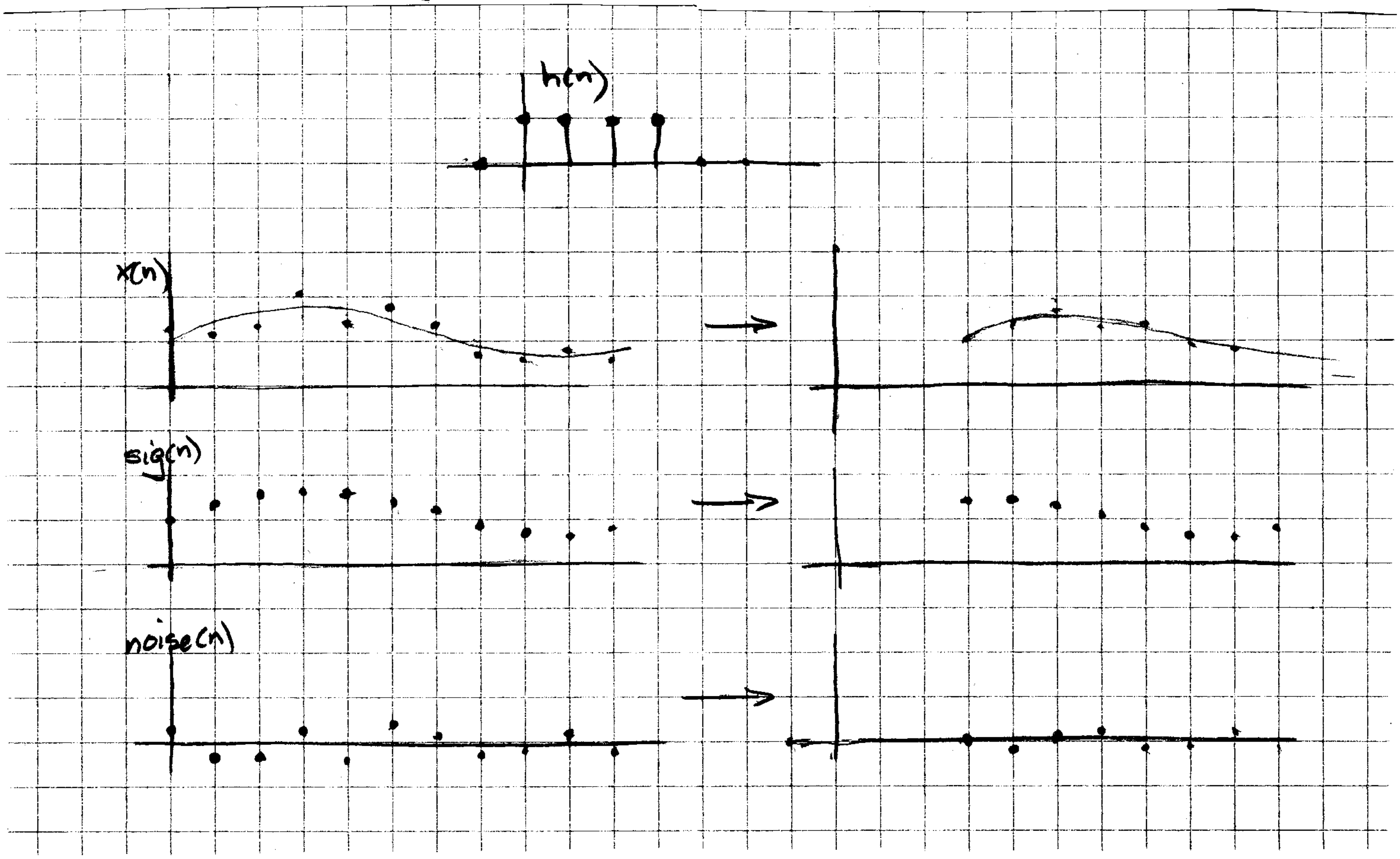
mult, add, to give $\boxed{}$ $= y(n)$

"flip, slide and integrate"

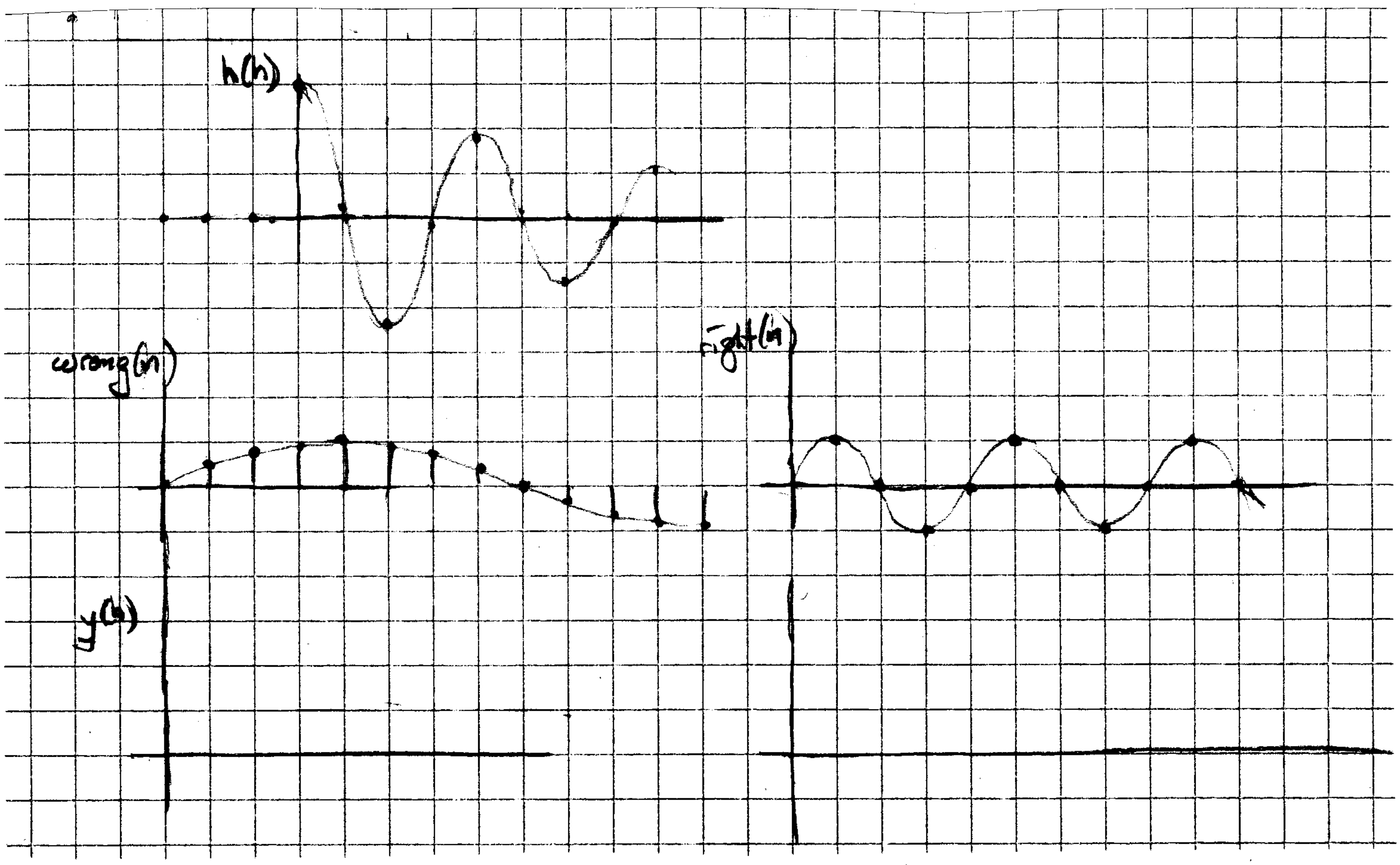
- Clearly, the convolution summation can be written as a program. If so, it's DSP.

- **Example** LPF to reduce high frequency measurement noise

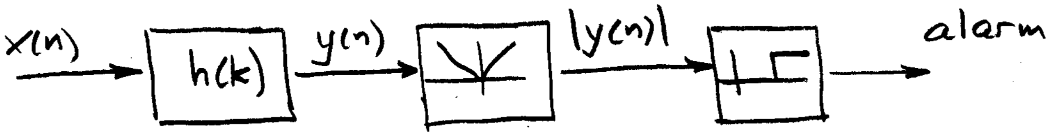
  Try a "moving window averager"



• **Example** BPF response to different frequencies

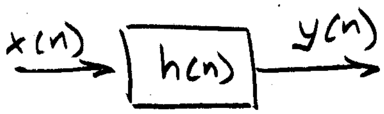


One possible use — tone coded control

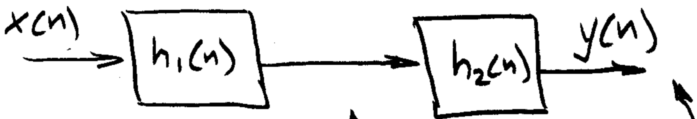


- Commutative property.

We saw that $x(n) \circledast h(n) = h(n) \circledast x(n)$

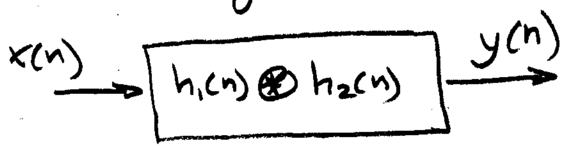


- Cascade connection (no loading)

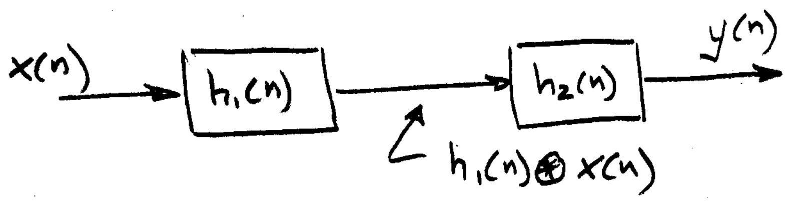


If $x(n) = \delta(n)$ then is $h_1(n)$ and $y(n) = h_2(n) \circledast h_1(n)$

Therefore it's equivalent to $h(n) = h_1(n) \circledast h_2(n)$



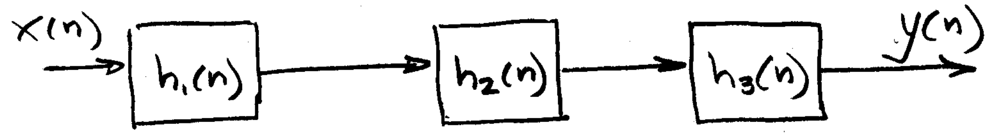
- Associative property

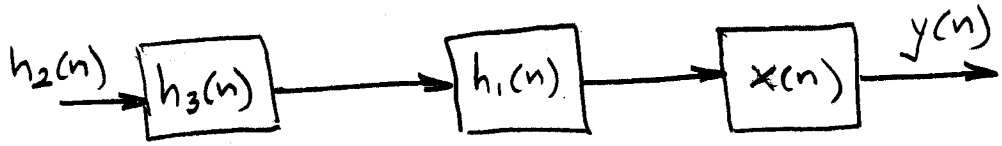


$$y(n) = h(n) \circledast x(n) = \left(h_2(n) \circledast h_1(n) \right) \circledast x(n)$$

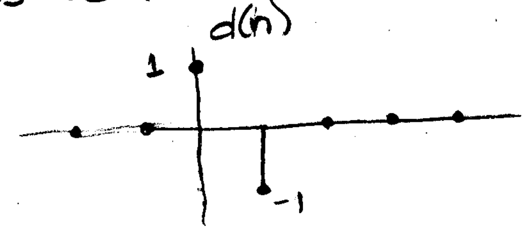
$$= h_2(n) \circledast \left(h_1(n) \circledast x(n) \right)$$

- Combined associative and commutative properties

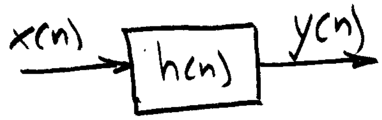


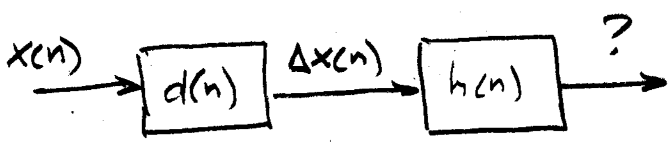


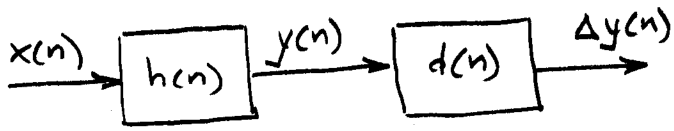
- The first difference operator can be viewed as a filter with unit pulse response



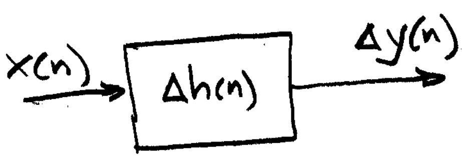
This gives an easy demonstration of the difference property

If 





$$\Delta y(n) = h(n) \circledast \Delta x(n)$$



$$\Delta y(n) = \Delta h(n) \circledast x(n)$$