Notes 3: Discrete-time LTI Systems

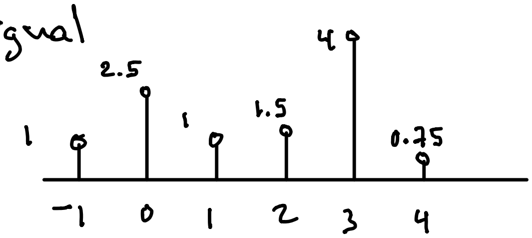
Objectives: You should be able to:

- Understand and explain why LTI systems are described by their impulse response.
- calculate a DT system output signal from the input and impulse response.
- explain / interpret the convolution operator
- use the linearity of convolution to simplify system output calculations
- translate input-output representations of LTI systems into an impulse response
- find the impulse response of a system from a block diagram

main

DT signals are sums of scaled δ-functions

We saw this earlier, but if you have a DT signal



then we can write it as

$$x[n] = 1\cdot\delta[n+1] + 2.5\,\delta[n] + 1\cdot\delta[n-1] + 1.5\,\delta[n-2] + 4\,\delta[n-3] + 0.75\,\delta[n-4]$$

More generally, as

$$\sum_{k=-\infty}^{\infty} x[k]\,\delta[n-k]$$

Ex

Examples:

Unit step $u[n] = \sum_{k=0}^{\infty} \delta[n-k]$

Complex exponential

$$e^{-j\omega n} = \sum_{k=-\infty}^{\infty} e^{-j\omega k}\,\delta[n-k]$$

Main

Essentially $\delta[n-k]$ picks out the k^{th} element of the signal.

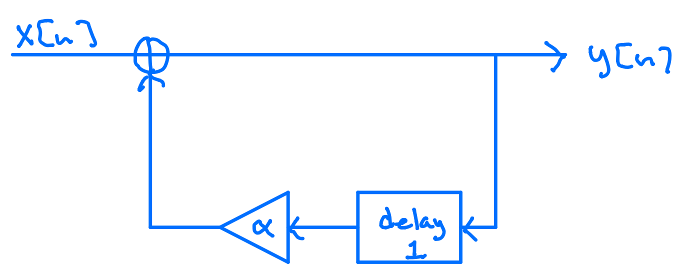
What we want to do is write the output of a DT LTI system using this representation of the signal.

!!!

The math involved in what we will be doing next involves some manipulation of summation notation and changing limits around. This requires some practice if you are not used to it. As always, DRAW A PICTURE and sketch the functions so you have a sense of what is going on.

Ex.

Let's revisit our example of the feedback system:

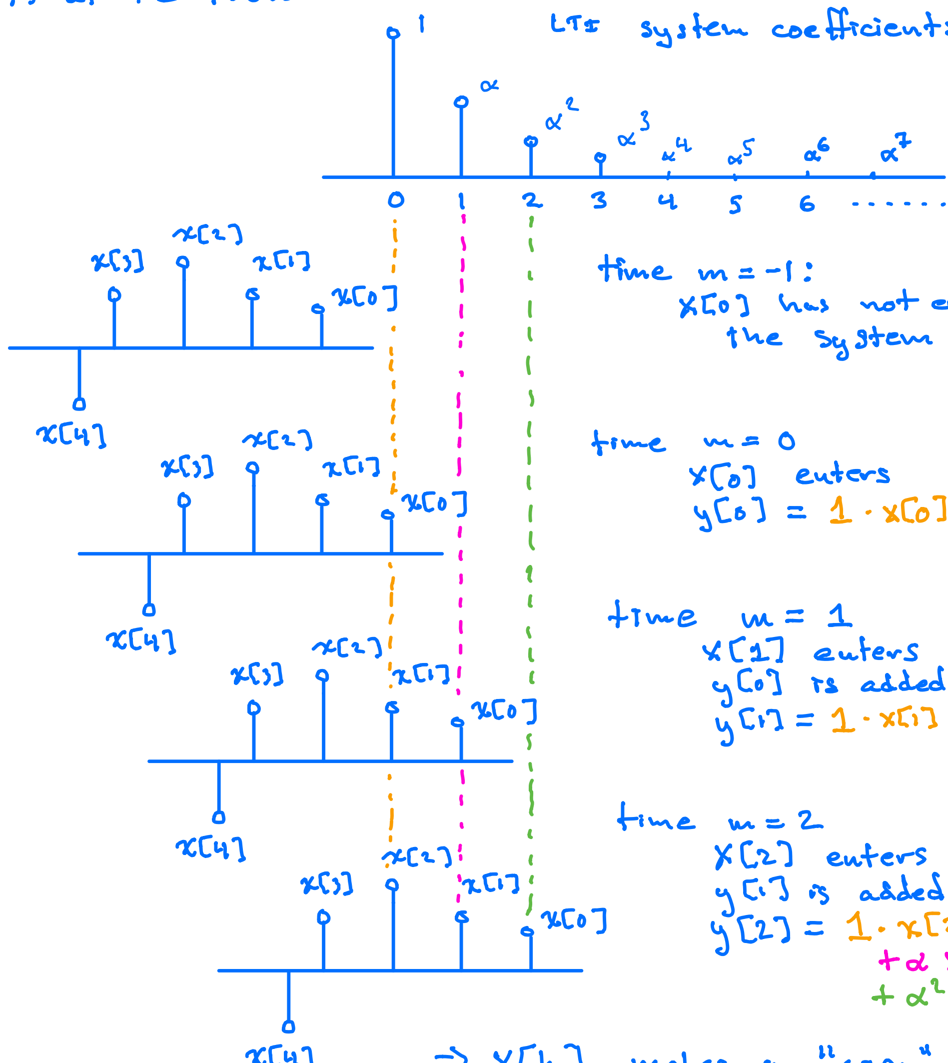


$$y[n] + \alpha\, y[n-1] = x[n]$$

We saw before that this can be written as a weighted accumulator:

$$y[n] = x[n] + \alpha x[n-1] + \alpha^2 x[n-2] + \dots + \alpha^n x[0]$$

How can we visualize what is going on? Picture $x[n]$ going <u>into</u> the system: this means $x[0]$ is at the front ...



time $n = -1$:
$x[0]$ has not entered the system

time $n = 0$
$x[0]$ enters
$y[0] = 1 \cdot x[0]$

time $n = 1$
$x[1]$ enters
$y[0]$ is added
$y[1] = 1 \cdot x[1] + \alpha x[0]$

time $n = 2$
$x[2]$ enters
$y[1]$ is added
$y[2] = 1 \cdot x[2] + \alpha x[1] + \alpha^2 x[0]$

$\Rightarrow$ $x[k]$ makes a "copy" of the coefficients delayed by k

Main

What does this picture tell us? It shows that we can think of an LTI system as being a set of coefficients and to find the output we slide the input signal across, multiply by the corresponding coefficients and then add them up.

Since we are sliding the input signal from left to right, we have to draw it backwards/flipped so that $x[0]$ hits the system first, followed by $x[1]$, etc..

Not

We often will call an LTI system a filter — this is the term used more often in signal processing & communications. For a DT filter, the coefficients are sometimes called taps. This terminology comes from how these systems were implemented in hardware.

Main

Important points:

- The signal is not (physically) "flipped" — this is just how we draw it
- Each input $x[k]$ makes a weighted copy of the coefficients starting at time k

!!!

This "flip-and-slide" operation might take some getting used to. If you can write down the picture it can help you *a lot* when trying to do the calculations. There are a few ways to practice / build up intuition.

1) Hands-on: draw $x[n]$ and $x[-n]$ on two small pieces of paper. Draw the filter coefficients on another. Practice sliding $x[-n]$ over the filter & writing down the output as you go along.

2) Use the DT convolution demo from SP First to visualize the convolution operation.

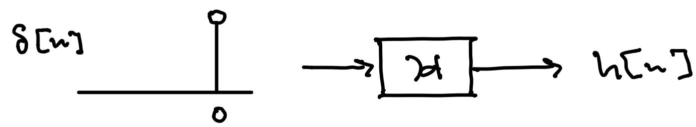
3) Practice the analytical calculations (the algebra) to make sure you have the indices correct.

All three can help & they help in different ways.

Main

Question: can <u>all</u> LTI systems be represented as a set of coefficients?

Yes. To see why, look at the system response to the unit impulse $\delta[n]$:



Call the output $h[n]$.

Def

Def: The <u>impulse response</u> of an LTI system is the output signal when $x[n] = \delta[n]$. Similarly, for CT signals it is the output of the system with input $\delta(t)$.

Main

Now, since the system is time-invariant, the output with $x[n] = \delta[n-k]$ is $h[n-k]$.

Because the system is linear, the output with $x[n] = a_{k_1}\delta[n-k_1] + a_{k_2}\delta[n-k_2]$ is $a_{k_1} h[n-k_1] + a_{k_2} h[n-k_2]$

Getting fancier, with input

$$x[n] = \sum_{k=-\infty}^{\infty} a_k \, \delta[n-k] \qquad (*)$$

the output is

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

Since $x[n]$ can always be written in the form $(*)$:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\,\delta[n-k]$$

the output of an LTI system is

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

Def

The formula in $(\star)$ is called the <u>discrete convolution</u> of $x[n]$ and $h[n]$.

Phil

The really cool thing about this little derivation is that it implies that everything we need to know about an LTI system is contained in the impulse response.

So to check the various system properties we saw earlier, we need to <u>translate</u> those <u>system</u> properties into <u>signal</u> properties of the impulse response $h[n]$.

Main

The convolution theorem (DT version): The output of an LTI system with impulse response $h[n]$ is given by the convolution of the input with $h[n]$:

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

Not

Convolutions are a major part of this class. We use a notation $*$ (an asterisk) to write the convolution:

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

OR

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

OR

$$y = x * h$$

One little point of possible confusion is that the formula with the summation sign is a formula for calculating y at time n, so:

$$y[10] = \sum_{k=-\infty}^{\infty} x[k]\,h[10-k]$$

The $*$ notation is for the <u>whole signal</u>, so for example, $y[10] = x[10] * h[10]$ <u>does NOT</u> make sense, but $y[10] = (x*h)[10]$ <u>does</u> make sense.

!!! This confusion where you might want to write $y[5] = x[5] * h[5]$ is something to avoid! Shorthand notation like writing $*$ for convolution is convenient if you understand what it's saying/summarizing.

Main

Writing $y = x * h$ for convolution raises an interesting question:

Does $x * h = h * x$?

To answer that we have to unpack the $*$ notation:

$$\sum_{k=-\infty}^{\infty} x[k]\, h[n-k] \stackrel{?}{=} \sum_{m=-\infty}^{\infty} h[m]\, x[n-m]$$

What if we substitute

$m = n-k$?

We have to check the range of the sum:

k:	...	-3	-2	-1	0	1	2	3	...
$m = n-k$:	...	$n+3$	$n+2$	$n+1$	n	$n-1$	$n-2$	$n-3$	...

So as k goes from $-\infty$ to ∞, m goes from ∞ to $-\infty$. Looking at the first sum:

$$k \longrightarrow n-m$$
$$n-k \longrightarrow m$$

so $$\sum_{k=-\infty}^{\infty} x[k]\,h[n-k] = \sum_{m=\infty}^{-\infty} x[n-m]\,h[m]$$

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

since the summation direction doesn't matter

$$= (h * x)[n]$$

So:

$$\boxed{\begin{array}{c}\text{Convolution is a symmetric operator} \\ x * h = h * x\end{array}}$$

Interpreting the convolution

If we look at the two ways of writing the convolution we can get different insights:

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

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

Formula (A) says that the output signal is the sum of scaled and shifted impulse responses.

- this is the "copying" operation from the picture
- you can think of $h[n]$ as like the sound of ringing a bell. The signal value $x[k]$ hits the bell starting at time k.
- Some people view the formula as "flipping the impulse response and sliding it across x." While mathematically it does look like that, <u>physically</u> the system is not flipping anything. So I would recommend against thinking about it this way.

Formula (B) corresponds to our picture view. The input enters the system from the left which means that we have to <u>draw</u> it flipped. The output at time n is calculated by multiplying the filter taps by the signal moved to the right by n time steps:

$$\begin{tikzcd}
x[k] \arrow[r, "\text{``flipping'' to draw it}"] \arrow[d, "\text{advancing } n \text{ steps forward}"'] & x[-k] \arrow[d] \\
x[k+n] \arrow[r] & x[-k+n]
\end{tikzcd}$$

Examples and calculations

If everything we want to know is in the impulse response (which we can think of as a signal), what about simple systems we saw earlier?

- Delay / advance by k

$$x[n] \longrightarrow \boxed{\begin{matrix}\text{delay}\\ k\end{matrix}} \longrightarrow y[n] = x[n-k]$$

$$h[n] = \delta[n-k]$$

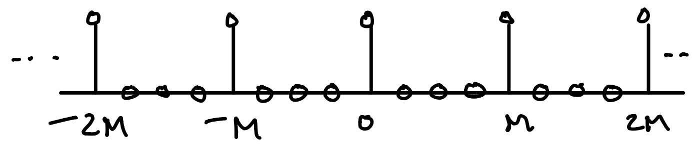
- Scaling by α

$$x[n] \longrightarrow \boxed{\alpha} \longrightarrow y[n] = \alpha\, x[n]$$

$$h[n] = \alpha\, \delta[n]$$

- Impulse train:

$$h[n] = \sum_{k=-\infty}^{\infty} \delta[n - kM]$$



What does convolving with $h[n]$ do? It puts a copy of $x[n]$ delayed by multiples of M.

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

Convolution and linearity:

Since the system is computing "convolution with h" and we already know the system is linear, we have:

$$(h * (a_1 x_1 + a_2 x_2)) = a_1 (h * x_1) + a_2 (h * x_2)$$

We can see this from the summation too:

$$\sum_{k=-\infty}^{\infty} h[k] \left(a_1 x_1[n-k] + a_2 x_2[n-k] \right)$$

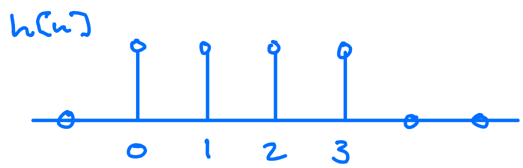
$$= a_1 \sum_{k=-\infty}^{\infty} h[k]\, x_1[n-k] + a_2 \sum_{k=-\infty}^{\infty} h[k]\, x_2[n-k]$$

Ex

Unfortunately, as with many things in life, one of the best ways to get better at convolutions is to do them.

Example: Suppose $h[n] = u[n] - u[n-4]$, what is the output when $x[n] = \alpha^n$ for $|\alpha| < 1$?

First thing to do: draw a picture. What is $h[n]$ and what is the system doing?



Why? $u[n]$ is 1 from $n=0$ on
$u[n-4]$ is 1 from $n=4$ on

so $x[n]$ is 1 from $n=0$ to $n=3$.

So $h[n]$ just adds up 4 input points in a row:

$$h[n] = \delta[n] + \delta[n-1] + \delta[n-2] + \delta[n-3]$$

$$\Rightarrow y[n] = x[n] + x[n-1] + x[n-2] + x[n-3]$$

$$= \alpha^n + \alpha^{n-1} + \alpha^{n-2} + \alpha^{n-3}$$

$$= \alpha^{n-3}(\alpha^3 + \alpha^2 + \alpha + 1)$$

Example: In this example we will make both $x[n]$ and $h[n]$ finite length.

Not

When talking about DT filters (systems) people often make a distinction between

finite impulse response (FIR)

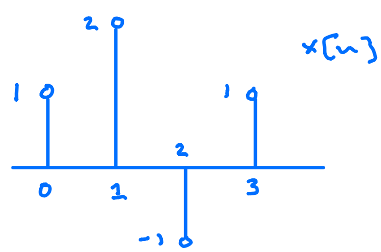
and infinite impulse response (IIR)

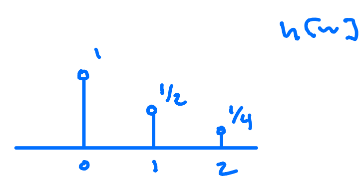
filters. FIR filters have $h[n]$ bounded in time, so there are N_{min} and N_{max} such that $h[n] = 0$ for $n < N_{min}$ and $n > N_{max}$.
IIR filters have infinite length in one (or both) directions

Suppose $h[n] = \delta[n] + \frac{1}{2}\delta[n-1] + \frac{1}{4}\delta[n-2]$

and $x[n] = \delta[n] + 2\delta[n-1] - \delta[n-2] + \delta[n-3]$

Drawing them out:





what is $y[n] = (x * h)[n]$?

Add up scaled and shifted versions of $h[n]$:

$$y[n] = \left(\delta[n] + \tfrac{1}{2}\delta[n-1] + \tfrac{1}{4}\delta[n-2]\right)$$

$x[0]\,h[n]$

$$+\,2\left(\delta[n-1] + \tfrac{1}{2}\delta[n-2] + \tfrac{1}{4}\delta[n-3]\right)$$

$x[1]\,h[n-1]$

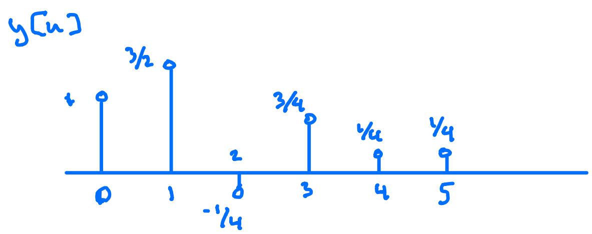
$$-\left(\delta[n-2] + \tfrac{1}{2}\delta[n-3] + \tfrac{1}{4}\delta[n-4]\right)$$

$x[2]\,h[n-2]$

$$+\left(\delta[n-3] + \tfrac{1}{2}\delta[n-4] + \tfrac{1}{4}\delta[n-5]\right)$$

$x[3]\,h[n-3]$

$$= \delta[n] + \tfrac{3}{2}\delta[n-1] - \tfrac{1}{4}\delta[n-2] + \tfrac{3}{4}\delta[n-3] + \tfrac{1}{4}\delta[n-4] + \tfrac{1}{4}\delta[n-5]$$



Main

This is one way to find the output of a system: write out scaled and shifted versions of the impulse response. Because convolution is symmetric, sometimes it's easier to use

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

and write scaled and shifted versions of the input signal. For the previous example:

$$y[n] = \left(\delta[n] + 2\delta[n-1] - \delta[n-2] + \delta[n-3]\right)$$
$$+\tfrac{1}{2}\left(\delta[n-1] + 2\delta[n-2] - \delta[n-3] + \delta[n-4]\right)$$
$$+\tfrac{1}{4}\left(\delta[n-2] + 2\delta[n-3] - \delta[n-4] + \delta[n-5]\right)$$

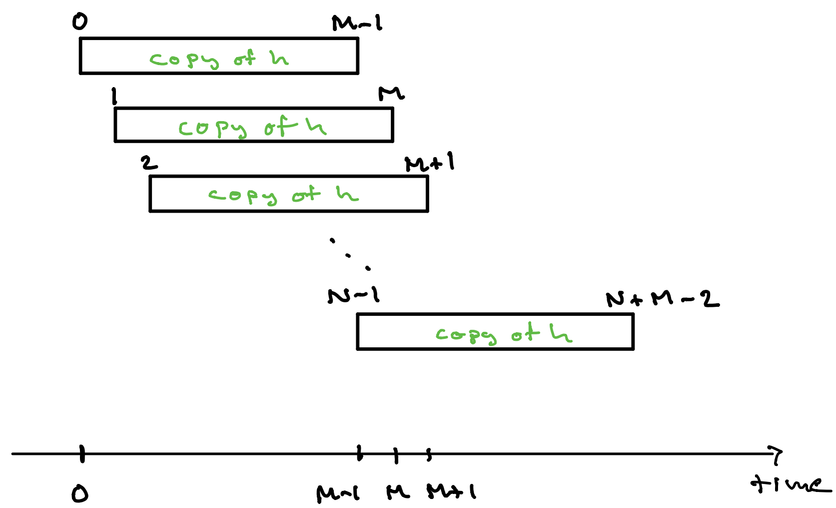
We would get the same answer.

This approach works ok when x and h are both finite length and pretty short.

A couple of notes:

- What is the length of $y[n]$? Think of the copies of $h[n]$: they are getting shifted by delays corresponding to each time step in $x[n]$.

  If $x[n]$ has length N and $h[n]$ has length M } starting at $n=0$



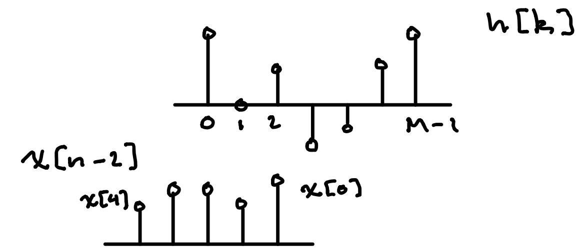
  Then $y[n]$ is nonzero starting at $n=0$ up to $n=N+M-2$, so it has length $N+M-1$

  This holds more generally: if $x[n]$ is length N and $h[n]$ is length M then $y[n]$ is length $N+M-1$.

- When doing convolutions for DT signals we can think of the output as having up to 3 "phases." This is easiest to think about using the flip and slide visualization:

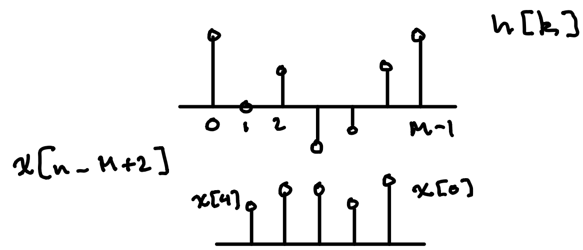
<u>Phase 1</u>

partial overlap "entering"



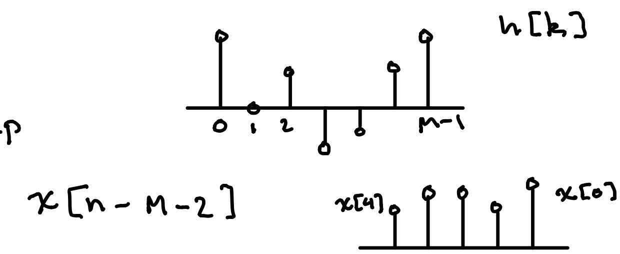
<u>Phase 2</u>

total overlap



<u>Phase 3</u>

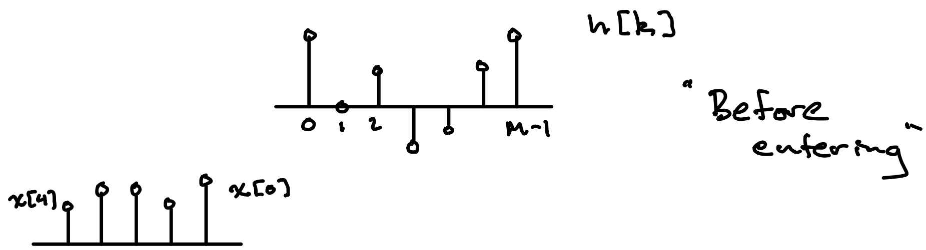
Partial overlap "exiting"

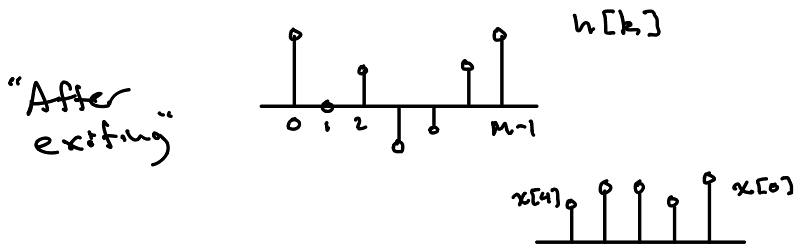


Mathematically this corresponds to having the general convolution formula going into one of 3 special cases where the limits on the sum (e.g. $k=-\infty$ to ∞) get simplified.

Some examples will help.

In the book they identify 5 different intervals (see example 2.4) — the two additional ones correspond to when the shifted version of $x[n]$ and $h[n]$ don't intersect at all:





Suppose $x[n]$ is nonzero only for

$$n = N_{min},\ N_{min}+1,\ \ldots,\ N_{max}$$

$h[n]$ is nonzero only for

$$n = M_{min},\ M_{min}+1,\ \ldots\ M_{max}$$

Then $x[n]$ has length $N = N_{max} - N_{min} + 1$

$h[n]$ has length $M = M_{max} - M_{min} + 1$

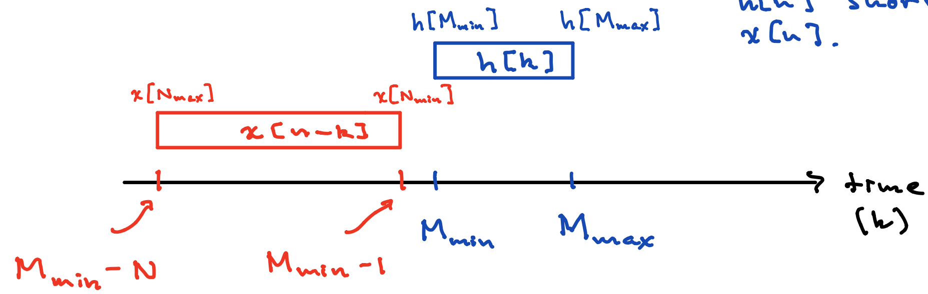
so $y[n]$ has length

$$N_{max} + M_{max} - (N_{min} + M_{min}) + 1.$$

What are the 3 (or 5) phases to calculate $y[n]$?

Right before entering:

Note: here we're making $h[n]$ shorter than $x[n]$.



$x[n-k]$ (as a signal over k with shift n) doesn't overlap with $h[k]$ until the front end of the flipped signal hits M_{min}.

We have to figure out what value of n that corresponds to:

$h[k]$ is nonzero from $k = N_{min}$ to N_{max}

$h[-k]$ is nonzero from $k = -N_{max}$ to $-N_{min}$

$h[n-k]$ is nonzero from $k = n - N_{max}$ to $n - N_{min}$

so

$$n - N_{min} = M_{min} - 1$$

or

$$n = M_{min} + N_{min} - 1$$

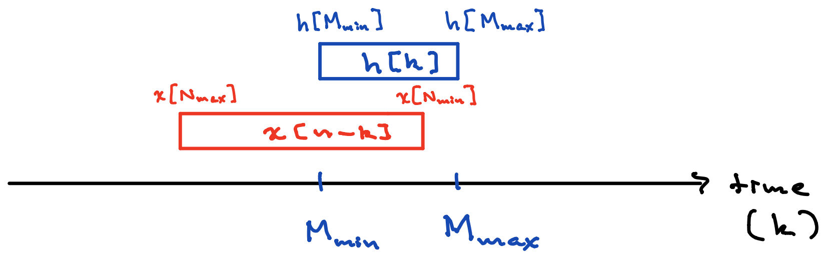
That means

$$y[n] = 0 \quad \text{for} \quad n = -\infty, \dots M_{min} + N_{min} - 1$$

Right as x is entering the filter, the time is

$$n = M_{min} + N_{min}.$$

Then we're in phase 1 until the front of $x[n-k]$ hits M_{max}:



that happens when

$$n - N_{min} = M_{max}$$

$$n = M_{max} + N_{min}$$

In this range:

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

$$= \sum_{k=M_{min}}^{M_{max}} h[k]\, x[n-k]$$

$$= \sum_{k=M_{min}}^{n-N_{min}} h[k]\, x[n-k]$$

So $$y[n] = \sum_{k=M_{min}}^{n-N_{min}} h[k]\, x[n-k]$$

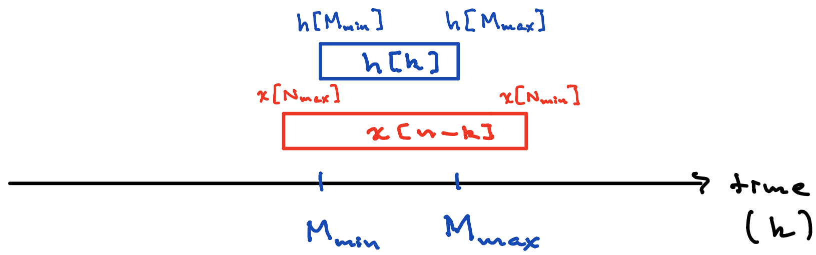
for $n = M_{min} + N_{min}$

to $n = M_{max} + N_{min} - 1$

Then we enter the overlap phase at time

$$n = M_{max} + N_{min}$$

This continues until the back end of x passes M_{min}:



Remember that $x[n-k]$ is nonzero from $k = n - N_{max}$ to $k = n - N_{min}$.

So the back of $x[n-k]$ hits M_{min} when

$$n - N_{max} = M_{min}$$

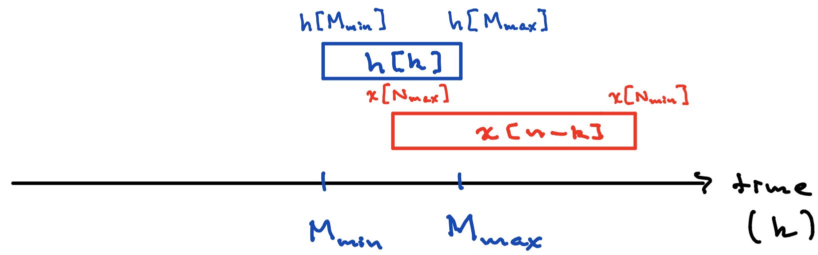
$$n = M_{min} + N_{max}$$

In this range we only sum over $k =$ from M_{min} to M_{max}:

$$y[n] = \sum_{k=M_{min}}^{M_{max}} h[k]\, x[n-k]$$

for $n = M_{max} + N_{min}$ to $n = M_{min} + N_{max}$

Then x starts to exit the filter:



This exiting phase runs from $n = M_{min} + N_{max} + 1$

to

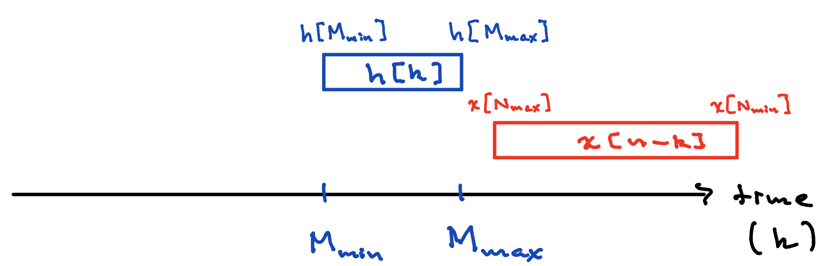
$$n - N_{max} = M_{max}$$

$$n = M_{max} + N_{max}$$

so

$$y[n] = \sum_{k=n-N_{max}}^{M_{max}} h[k]\, x[n-k] \quad \text{for } n = M_{min} + N_{max} + 1 \text{ to } n = M_{max} + N_{max}$$

Then x no longer overlaps the filter:



$$y[n] = 0 \quad \text{for } n > M_{max} + N_{max}$$

Putting it all together:

$$y[n] = \begin{cases} 0 & n < M_{min} + N_{min} \\ \displaystyle\sum_{k=M_{min}}^{n-N_{min}} h[k]\, x[n-k] & M_{min} + N_{min} \le n < M_{max} + N_{min} \\ \displaystyle\sum_{k=M_{min}}^{M_{max}} h[k]\, x[n-k] & M_{max} + N_{min} \le n \le M_{min} + N_{max} \\ \displaystyle\sum_{k=n-N_{max}}^{M_{max}} h[k]\, x[n-k] & M_{min} + N_{max} < n \le M_{max} + N_{max} \\ 0 & n > M_{max} + N_{max} \end{cases}$$

Try

We did all of this for the case where $h[n]$ is shorter than $x[n]$. What happens if $h[n]$ is longer than $x[n]$? You can either rederive everything from scratch (good for intuition / visualization) or exploit the properties of convolution...

Main

So for a finite $x[n]$ through an FIR $h[n]$ we can calculate the output from the definition or by using this 3 (or 5) phase breakdown.

Ex

Example (Oppenheim & Willsky Example 2.4) (slightly modified)

Suppose

$$h[n] = \begin{cases} 1 & 0 \le n \le 4 \\ 0 & \text{otherwise} \end{cases}$$

$$x[n] = \begin{cases} \alpha^n & 0 \le n \le 6 \\ 0 & \text{otherwise} \end{cases}$$

Find $y[n] = (x * h)[n]$.

First: $N_{min} = 0$ $\quad N_{max} = 6$ $\quad N = 7$

$M_{min} = 0$ $\quad M_{max} = 4$ $\quad M = 5$

$y[n]$ will have length $N + M - 1 = 11$

from $n = N_{min} + M_{min} = 0$

to $n = N_{max} + M_{max} = 10$

Now apply the formula:

$$y[n] = \begin{cases} 0 & n < 0 \\ \sum_{k=0}^{n} 1 \, \alpha^{n-k} & 0 \le n < 4 \\ \sum_{k=0}^{4} 1 \cdot \alpha^{n-k} & 4 \le n \le 6 \\ \sum_{k=n-6}^{4} 1 \cdot \alpha^{n-k} & 6 < n \le 10 \\ 0 & n > 10 \end{cases}$$

We're not quite done yet: have to do some algebra/calculus!

set $r = n-k$

$$\sum_{k=0}^{n} \alpha^{n-k} = \sum_{r=n}^{0} \alpha^{r} = \sum_{r=0}^{n} \alpha^{r}$$

$$= \sum_{r=0}^{\infty} \alpha^{r} - \sum_{r=n+1}^{\infty} \alpha^{r}$$

$$= \frac{1}{1-\alpha} - \alpha^{n+1} \sum_{r=0}^{\infty} \alpha^{r}$$

$$= \frac{1-\alpha^{n+1}}{1-\alpha}$$

$$\sum_{k=0}^{4} \alpha^{n-k} = \alpha^{n} + \alpha^{n-1} + \alpha^{n-2} + \alpha^{n-3} + \alpha^{n-4}$$

$$= \alpha^{n-4} \sum_{k=0}^{4} \alpha^{k}$$

using what we showed just above

$$= \alpha^{n-4} \left(\frac{1-\alpha^{4+1}}{1-\alpha} \right)$$

$$= \frac{\alpha^{n-4} - \alpha^{n+1}}{1-\alpha}$$

$r = k-n+6$

$$\sum_{k=n-6}^{4} \alpha^{n-k} = \sum_{r=0}^{10-n} \alpha^{6-r}$$

$$= \alpha^{6} + \alpha^{5} + \cdots + \alpha^{-4+n}$$

$$= \alpha^{n-4} \left(1 + \alpha + \cdots + \alpha^{10-n} \right)$$

$$= \alpha^{n-4} \sum_{k=0}^{10-n} \alpha^{k}$$

$$= \alpha^{n-4}\left(\frac{1-\alpha^{11-n}}{1-\alpha}\right)$$

$$= \frac{\alpha^{n-4}-\alpha^{7}}{1-\alpha}$$

Try

Try out using the general formula on a few examples including the problems <u>not</u> assigned as homework.

Main

Since we are going to be seeing a lot of the (truncated) geometric series in this class it helps to have a few formulas handy (but also know how to rederive them):

$$\sum_{k=0}^{\infty} \alpha^k = \frac{1}{1-\alpha}$$

$$\sum_{k=r}^{\infty} \alpha^k = \alpha \sum_{k=0}^{\infty} \alpha^k = \frac{\alpha^r}{1-\alpha}$$

$$\sum_{k=0}^{s} \alpha^k = \sum_{k=0}^{\infty} \alpha^k - \sum_{k=s+1}^{\infty} \alpha^k = \frac{1-\alpha^{s+1}}{1-\alpha}$$

$$\sum_{k=r}^{s} \alpha^k = \sum_{k=r}^{\infty} \alpha^k - \sum_{k=s+1}^{\infty} \alpha^k$$

$$= \frac{\alpha^r - \alpha^{s+1}}{1-\alpha}$$

Translating DT system properties into impulse response properties

Because the impulse response tells us everything we need to know about LTI systems, we should be able to figure out if different system properties hold based on the impulse response alone.

Unfortunately, we are going to have to wait until we do z-transforms to cover stability and invertiblity...

But causality is easy! LTI systems are

- memoryless if $h[n] = \alpha \delta[n]$ for some α.
- causal if $h[n] = \sum_{k=0}^{\infty} \alpha_k \delta[k]$
- anticausal if $h[n] = \sum_{k=-\infty}^{0} \alpha_k \delta[k]$
- noncausal if $$h[n] = \sum_{k=M_{min}}^{M_{max}} \alpha_k \delta[k]$$

  where $M_{min} < 0$ and $M_{max} > 0$.