



Course material

Course:

Digital Signal Processing II

Video:

2.1.1.b Convolution

Concepts (extracted from automatically generated subtitles):

Impulse response. Power of n times. Exponentially decaying sequence h . Output of a filter. Arbitrary sequence x . Inner product. Convolution of sequences x . Times delta. Delta function. Three-point sequence. Current sample n . Components of the signal. Minus infinity. Linear combination of delayed delta functions. Unit step.



[to video sequence search](#)
(within Digital Signal Processing II.)



[to video](#)

Center for Digital Education. More educational support material here:

<https://exoset.epfl.ch/booc/>

Impulse response

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

Fundamental result: impulse response fully characterizes the LTI system!

The impulse response is the output of a filter when the input is the delta function. A fundamental result states that the impulse response fully characterizes the behavior of an LTI system. Let's see why that is so.

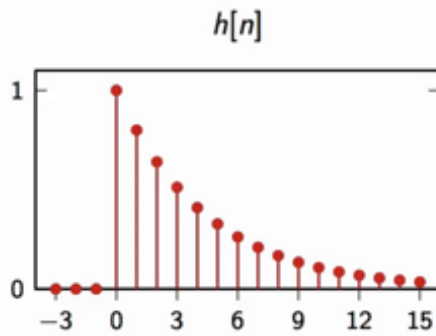
notes

summary

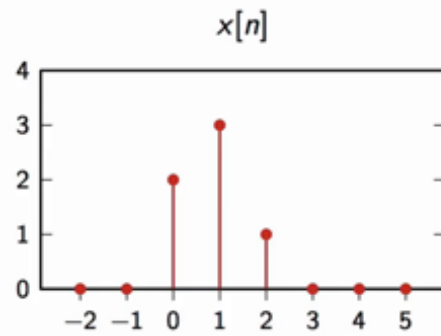
0m 0s



Example



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



$$x[n] = \begin{cases} 2 & n = 0 \\ 3 & n = 1 \\ 1 & n = 2 \\ 0 & \text{otherwise} \end{cases}$$

Assume that we have a filter and we can measure its impulse response by inputting a delta function, and it turns out that the impulse response is an exponentially decaying sequence h of n equal to α to the power of n times the unit step. Now we want to use the same filter to filter an arbitrary sequence x of n , that in this example is simply a three-point sequence that is equal to 2 for n equal to 0,

notes

summary

0m 11s



Example

- ▶ $x[n] = 2\delta[n] + 3\delta[n-1] + \delta[n-2]$
- ▶ we know the impulse response $h[n] = \mathcal{H}\{\delta[n]\}$;
- ▶ compute $y[n] = \mathcal{H}\{x[n]\}$ exploiting linearity and time-invariance

is equal to 3 for n equal to 1, and is equal to 1 for n equal to 2, and is 0 everywhere else. So we can always write our sequence as a linear combination of delayed delta functions.

notes

summary

0m 37s



Example

- ▶ $x[n] = 2\delta[n] + 3\delta[n-1] + \delta[n-2]$
- ▶ we know the impulse response $h[n] = \mathcal{H}\{\delta[n]\}$;
- ▶ compute $y[n] = \mathcal{H}\{x[n]\}$ exploiting linearity and time-invariance

So in particular, for our example, x of n is equal to 2 times delta of n plus 3 times delta of n minus 1 plus delta of n minus 2. Now we know the impulse response, so the response to the delta, and by exploiting linearity and time invariance, we can compute the response of the system to the input sequence x of n just by knowing the impulse response. Indeed, we apply the linear filter to the linear combination of deltas,

notes

summary

0m 49s



Example

$$y[n] = \mathcal{H}\{2\delta[n] + 3\delta[n-1] + \delta[n-2]\}$$

$$= 2\mathcal{H}\{\delta[n]\} + 3\mathcal{H}\{\delta[n-1]\} + \mathcal{H}\{\delta[n-2]\}$$

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

and by exploiting linearity first, we can split the operation of the filter over the three components of the signal,

notes

summary

1m 19s



Example

$$y[n] = \mathcal{H}\{2\delta[n] + 3\delta[n-1] + \delta[n-2]\}$$

$$= 2\mathcal{H}\{\delta[n]\} + 3\mathcal{H}\{\delta[n-1]\} + \mathcal{H}\{\delta[n-2]\}$$

$$= \underline{2h[n] + 3h[n-1] + h[n-2]}$$

and by exploiting time invariance, we just sum together appropriately scaled version of the impulse response delayed. We can look at this graphically, and we see that when we filter the first component, 2 times delta of n,

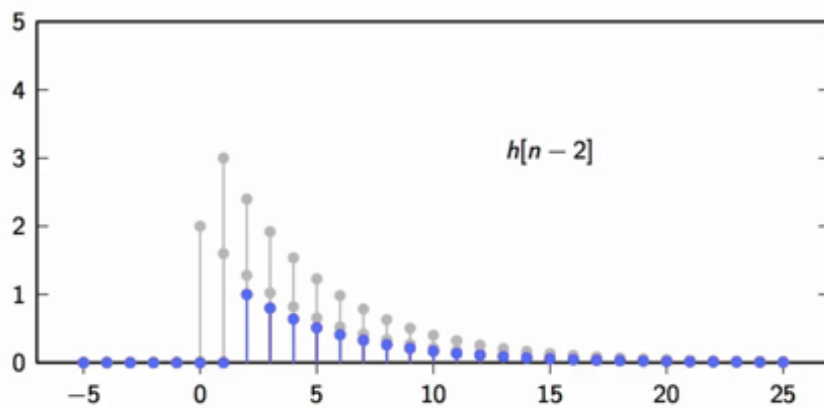
notes

summary

1m 25s



Example



we get 2 times h of n . The second component of the signal is 3 times δ of n minus 1, and this gives rise to $3h$ of n minus 1, which we plot on top of the other response. And finally, the last component is simply δ of n minus 2, which filter gives h of n minus 2. So now we have the three components, we sum them together by linearity, and we obtain the response of the system to our arbitrary input.

notes

summary

1m 40s



Convolution

We can always write (remember Module 3.2):

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

by linearity and time invariance:

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} x[k]h[n-k] \\ &= x[n] * h[n] \end{aligned}$$

In general, remember that we can always write x of n , a generic discrete time sequence, as the sum for k that goes from minus infinity to plus infinity of a sequence of time delayed deltas,

notes

summary

2m 13s



Convolution

We can always write (remember Module 3.2):

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

by linearity and time invariance:

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} x[k]h[n-k] \\ &= x[n] * h[n] \end{aligned}$$

scaled by the values of the sequence itself. So this probably seemed like a futile exercise in module 3.2, but now we see the usefulness of this representation,

notes

summary

2m 25s



Convolution

We can always write (remember Module 3.2):

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

by linearity and time invariance:

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} x[k]h[n-k] \\ &= x[n] * h[n] \end{aligned}$$

because by linearity and time invariance, we can express the output as the sum from k that goes from minus infinity to plus infinity of the values of the sequence times the impulse response time reversed and delayed by n .

notes

summary

2m 40s



Convolution

We can always write (remember Module 3.2):

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

by linearity and time invariance:

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} x[k]h[n-k] \\ &= x[n] * h[n] \end{aligned}$$

This sum here is so important in signal processing that it gets its own name, and it's called the convolution of sequences x of n and h of n . The convolution, which represents the output of a filter given its impulse response and an arbitrary input sequence x of n ,

notes

summary

2m 54s



Performing the convolution algorithmically

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

Ingredients:

- ▶ a sequence $x[m]$
- ▶ a second sequence $h[m]$

The recipe:

- ▶ time-reverse $h[m]$
- ▶ at each step n (from $-\infty$ to ∞):
 - center the time-reversed $h[m]$ in n (i.e. shift by $-n$)
 - compute the inner product

is actually an algorithmic formula to compute the output of the filter. The ingredients are an input sequence x of n and a second sequence h of n , and the recipe involves the following steps.

notes

summary

3m 10s



Performing the convolution algorithmically

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

Ingredients:

- ▶ a sequence $x[m]$
- ▶ a second sequence $h[m]$



The recipe:

- ▶ time-reverse $h[m]$
- ▶ at each step n (from $-\infty$ to ∞):
 - center the time-reversed $h[m]$ in n (i.e. shift by $-n$)
 - compute the inner product

First, we time reverse the impulse response, so we flip it in time.

notes

summary

3m 25s



Performing the convolution algorithmically

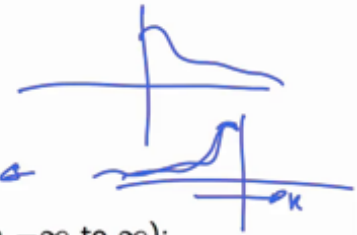
$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Ingredients:

- ▶ a sequence $x[m]$
- ▶ a second sequence $h[m]$

The recipe:

- ▶ time-reverse $h[m]$
- ▶ at each step n (from $-\infty$ to ∞):
 - center the time-reversed $h[m]$ in n (i.e. shift by $-n$)
 - compute the inner product



If it goes like this, then it will look like this. And at each step, from minus infinity to plus infinity, we center the time reversed impulse response in the current sample n . So we shift the time reversed impulse response by minus n , and then we compute the inner product between this shifted replica of the impulse response and the input sequence.

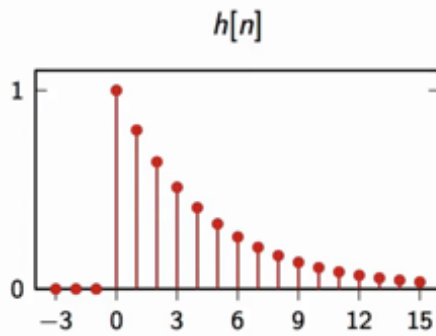
notes

summary

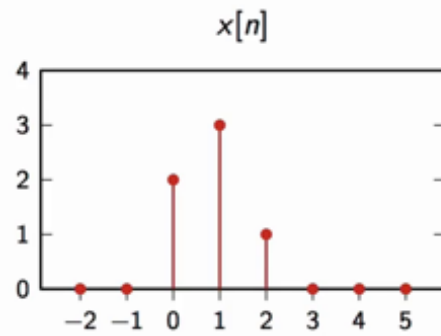
3m 30s



Same example, different perspective



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



$$x[n] = \begin{cases} 2 & n = 0 \\ 3 & n = 1 \\ 1 & n = 2 \\ 0 & \text{otherwise} \end{cases}$$

Let's look at this graphically using the same examples that we used before. So we have an impulse response, which is a decaying exponential sequence, and we have a three-point simple input sequence. We plot these three actors on a chart like this.

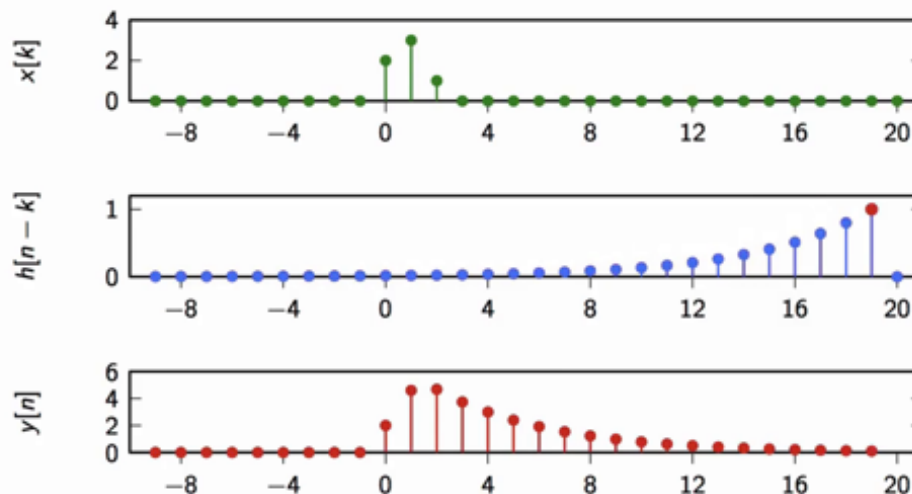
notes

summary

4m 1s



Convolution example



We have the input sequence on top. We have the time reversed and the delay impulse response on the second panel. And here we have the inner product between these two sequences. So at each step, as I said, we center the time reversed impulse response on the current sample. So we start at minus four and we compute the inner product. Since the input signal in our example is non-zero only between zero and two, up to zero, fundamentally nothing happens and the inner product is always zero. And we can see that it was zero for values before minus four, and it will continue to be zero until we hit zero. At which point we start to have an overlap between these two sequences and the inner product will not be zero. In particular, on the first step, it will be equal to this sample, which is equal to one, times this sample, which is equal to two. So the sum will be equal to two. We advance another step and then we see that the overlap involves two points now here and here. And we compute their products and their sum and we get the second point in our output sequence. Third step will finally involve all three points from the input sequence. We compute the product with the impulse response and the sum and we get our third output sample. And the process continues like so. Now, since the impulse response is an infinite sequence, one-sided infinite sequence, from now on, the inner product will always be non-zero and will have an output that will continue to be non-zero forever and ever. Finally, a few words on

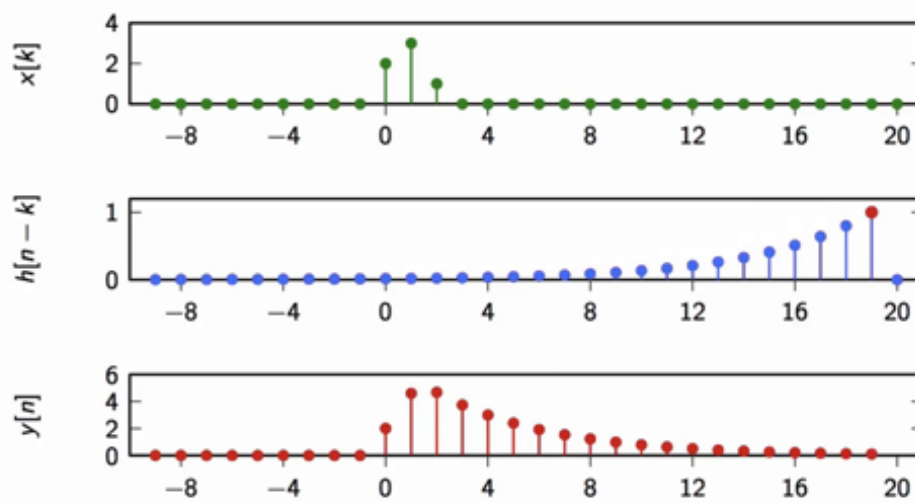
notes

summary

4m 14s



Convolution example



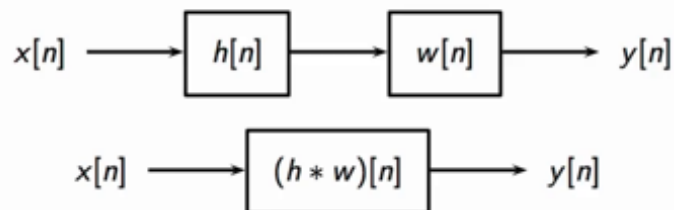
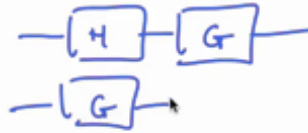
the convolution.

notes

summary

Convolution properties

- ▶ linearity and time invariance (by definition)
- ▶ commutativity: $(x * h)[n] = (h * x)[n]$
- ▶ associativity for absolutely- and square-summable sequences:
 $((x * h) * w)[n] = (x * (h * w))[n]$



The convolution is, of course, linear and time-invariant because it describes a linear and time-invariant operation. It is also commutative, which means that if you have two filters in cascade, you can safely invert their sequence and the result will not change.

notes

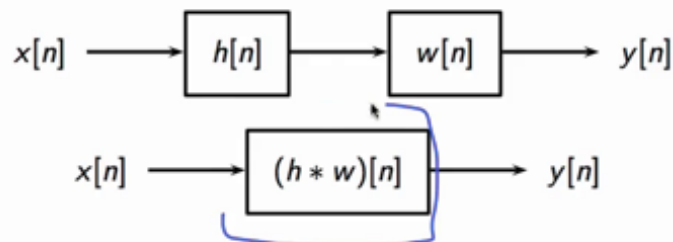
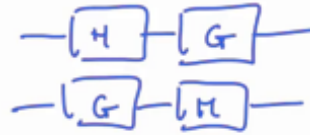
summary

6m 2s



Convolution properties

- ▶ linearity and time invariance (by definition)
- ▶ commutativity: $(x * h)[n] = (h * x)[n]$
- ▶ associativity for absolutely- and square-summable sequences:
 $((x * h) * w)[n] = (x * (h * w))[n]$



For absolutely and square-summable sequences, the convolution is also associative, which means that if you have a cascaded system, you can lump their effect into a single filter

notes

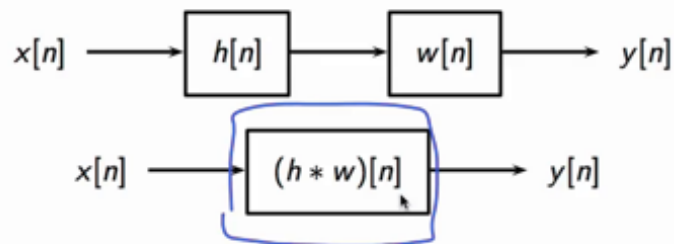
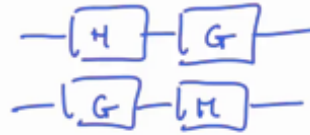
summary

6m 22s



Convolution properties

- ▶ linearity and time invariance (by definition)
- ▶ commutativity: $(x * h)[n] = (h * x)[n]$
- ▶ associativity for absolutely- and square-summable sequences:
 $((x * h) * w)[n] = (x * (h * w))[n]$



whose impulse response is the convolution of the individual impulse responses.

notes

summary

6m 35s

