

Course material

Course:

Digital Signal Processing II

Video:

2.2.2.a The z-transform

Concepts (extracted from automatically generated subtitles):

Z-transform. So-called constant coefficient difference equation. Most general linear time. Problem of polynomial multiplication. Second term. Function of a complex variable z . Invariant system. Finite support sequences. Given impulse response h . First coefficient. Minus λ times λ . Linear time. So-called frequency response. N output samples. Transfer function of the system.



[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/>

- ▶ ideal filters cannot be implemented
- ▶ what is the most general, realizable LTI transformation?
 - linearity: only sums and multiplications
 - time-invariance: only multiplications by constants
 - realizability: only finite number of past and future samples

We have seen in the previous lectures that ideal filters cannot be implemented

notes

summary

0m 0s



- ▶ ideal filters cannot be implemented
- ▶ what is the most general, realizable LTI transformation?
 - linearity: only sums and multiplications
 - time-invariance: only multiplications by constants
 - realizability: only finite number of past and future samples

and so a natural question is what is the most general linear time invariant transformation that we can implement in practice? So linearity implies that we can only use sums and multiplications

notes

summary

0m 1s



- ▶ ideal filters cannot be implemented
- ▶ what is the most general, realizable LTI transformation?
 - linearity: only sums and multiplications
 - time-invariance: only multiplications by constants
 - realizability: only finite number of past and future samples

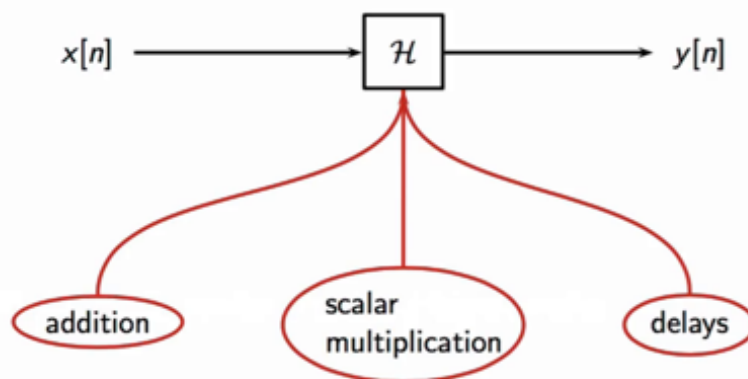
in our algorithm. Time invariance means that we can only multiply our quantities by constants and realisability means that in our algorithm we can only use a finite number of past or potentially future input and output samples. This is a graphical representation if you want what goes on

notes

summary

0m 13s





inside a linear time invariant system. The three fundamental ingredients are addition, scalar multiplication and delays. And mathematically we can fully describe

notes

summary

0m 34s



$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$$

- ▶ uses $M + 1$ input and N output values
- ▶ $a_0 = 1$ (otherwise renormalize)
- ▶ completely specified by $M + N + 1$ scalar coefficients

the inner workings of a linear time invariant system by a so-called constant coefficient difference equation that we see here in its most general form. It uses m plus 1 input samples

notes

summary

0m 46s



$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$$

- ▶ uses $M + 1$ input and N output values
- ▶ $a_0 = 1$ (otherwise renormalize)
- ▶ completely specified by $M + N + 1$ scalar coefficients

4

and n output samples. These are the limits in the summations here. The first coefficient in the linear combination of output samples is conventionally set to 1. If that is not the case we can renormalize the coefficient so that it is so. And the whole system is completely specified by m plus n plus 1 scalar coefficients. We can rearrange this equation in causal form and express the output y of n

notes

summary

1m 0s



$$y[n] = \sum_{k=0}^M b_k x[n-k] - \sum_{k=1}^N a_k y[n-k]$$

- causal formulation
- provides an *algorithm* to compute each output value
- how do we compute the frequency response? Well, it's the DTFT of...
- how do we compute the impulse response?

as a linear combination of past input samples weighed by the coefficients b_k plus a linear combination of past output samples weighed by scalar coefficient a_k . This causal formulation provides an

notes

summary

1m 32s



Constant-Coefficient Difference Equation



$$y[n] = \underbrace{\sum_{k=0}^M b_k x[n-k]}_{\text{causal}} - \underbrace{\sum_{k=1}^N a_k y[n-k]}_{\text{causal}}$$

- ▶ causal formulation
- ▶ provides an *algorithm* to compute each output value
- ▶ how do we compute the frequency response? Well, it's the DTFT of...
- ▶ how do we compute the impulse response?

5

algorithm that we can actually use to compute each

notes

summary

1m 49s



$$y[n] = \sum_{k=0}^M b_k x[n-k] - \sum_{k=1}^N a_k y[n-k]$$

- ▶ causal formulation
- ▶ provides an *algorithm* to compute each output value
- ▶ how do we compute the frequency response? Well, it's the DTFT of...
- ▶ how do we compute the impulse response?

5

output value. We just put the samples in and we run the machine. We have also seen that a linear time invariant system is fully characterized by its so-called frequency response. And the frequency response is very important because it gives us the properties of the system in the frequency domain. How do we compute the frequency response? Well it is the DTFT of the impulse response of the system.

notes

summary

1m 51s



$$y[n] = \sum_{k=0}^M b_k x[n-k] - \sum_{k=1}^N a_k y[n-k]$$

- ▶ causal formulation
- ▶ provides an *algorithm* to compute each output value
- ▶ how do we compute the frequency response? Well, it's the DTFT of...
- ▶ how do we compute the impulse response?

$$h[n] = (1 - \lambda) \lambda^n u[n]$$

So we shift the question to how do we compute the impulse response of a system given the constant coefficient difference equation that we have here. Now in the past we have explicitly computed the impulse response of simple systems like the moving average or the Leakey integrator. For instance the Leakey integrator has an impulse response that is equal to 1 minus lambda times lambda to the power of n times the step signal. But in general given a constant coefficient difference equation computed the impulse response explicitly is very complicated. So for this we need a new tool called the Z-transform. Before we introduce the Z-transform formally let's take a little detour that perhaps may appear a little esoteric at first but we will see the connection in a second. Let's consider the problem of

notes

summary

2m 16s



$$\begin{aligned}
 (1 + 3t + 2t^2)(2 + t - t^2 + 4t^3) &= 2 + t - t^2 + 4t^3 \\
 &\quad + 6t + 3t^2 - 3t^3 + 12t^4 \\
 &\quad + 4t^2 + 2t^3 - 2t^4 + 8t^5 \\
 &= 2 + 7t + 6t^2 + 3t^3 + 10t^4 + 8t^5
 \end{aligned}$$

6

polynomial multiplication. So suppose you have two polynomials like 1 plus 3t plus 2t square that multiplies 2 plus t minus t square plus 4t cube. And you remember from high school that the way you perform this multiplication is that you say for instance choose one polynomial, let's take the second term and you multiply this term by each of the monomials in the first term and you align them by

notes

summary

3m 9s



$$\begin{aligned}
 (1 + 3t + 2t^2)(2 + t - t^2 + 4t^3) &= 2 + t - t^2 + 4t^3 \\
 &\quad + 6t + 3t^2 - 3t^3 + 12t^4 \\
 &\quad + 4t^2 + 2t^3 - 2t^4 + 8t^5 \\
 &= 2 + 7t + 6t^2 + 3t^3 + 10t^4 + 8t^5
 \end{aligned}$$

6

corresponding powers of the free variable t .

notes

summary

3m 37s



6

notes

3m 38s



6

notes

3m 49s



Now try this...



$$\begin{aligned} [\dots 1 \ 3 \ 2 \ 0 \ 0 \ \dots] * [\dots 2 \ 1 \ -1 \ 4 \ 0 \ \dots] = \\ [\dots 2 \ 7 \ 6 \ 3 \ 10 \ 8 \ 0 \ \dots] \end{aligned}$$

7

Let's take two finite support sequences

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

summary

4m 11s



.....

.....

.....

.....

.....

Now try this...



$$\begin{array}{c} \dots \quad 1 \quad 3 \quad 2 \quad 0 \quad 0 \quad \dots \\ u=0 \end{array} * \begin{array}{c} \dots \quad 2 \quad 1 \quad -1 \quad 4 \quad 0 \quad \dots \\ \dots \quad 2 \quad 7 \quad 6 \quad 3 \quad 10 \quad 8 \quad 0 \quad \dots \end{array} =$$

7

where the non-zero coefficients, starting for n equal to 0 here, are equal to the coefficients of the two polynomials that we just saw. So we go from 1 plus 3t plus 2t square and this gives rise to a finite support

notes

summary

4m 14s



$$\begin{aligned}
 (1 + 3t + 2t^2)(2 + t - t^2 + 4t^3) &= 2 + t - t^2 + 4t^3 \\
 &\quad + 6t + 3t^2 - 3t^3 + 12t^4 \\
 &\quad + 4t^2 + 2t^3 - 2t^4 + 8t^5 \\
 &= 2 + 7t + 6t^2 + 3t^3 + 10t^4 + 8t^5
 \end{aligned}$$

6

sequence where the non-zero coefficients are 1, 3, 2. And we do the same for the second polynomial 2 plus t minus t square plus 4t cube and you would have the sequence 2, 1 minus 1, 4.

notes

summary

4m 27s





7



$$\begin{aligned}
 (1 + 3t + 2t^2)(2 + t - t^2 + 4t^3) &= 2 + t - t^2 + 4t^3 \\
 &\quad + 6t + 3t^2 - 3t^3 + 12t^4 \\
 &\quad + 4t^2 + 2t^3 - 2t^4 + 8t^5 \\
 &= 2 + 7t + 6t^2 + 3t^3 + 10t^4 + 8t^5
 \end{aligned}$$

6

So 2, 7, 6, 3, 10, 8. We go back and indeed it's 2, 7, 6, 3, 10, 8. So polynomial multiplication and convolution

notes

summary

4m 58s



Polynomial multiplication and convolution are the same



$$p(t) = p_0 + p_1 t + \dots + p_P t^P$$

$$q(t) = q_0 + q_1 t + \dots + q_Q t^Q$$

$$r(t) = p(t) \cdot q(t) = \sum_{n=0}^{P+Q} r_n t^n$$

$$r_n = \sum_{k=\max\{0, n-Q\}}^{\min\{n, P\}} p_k q_{n-k}, \quad 0 \leq n \leq P+Q$$

8

are actually the same. In the general case, given two polynomials

notes

summary

5m 8s



Polynomial multiplication and convolution are the same



$$p(t) = p_0 + p_1 t + \dots + p_P t^P$$

$$q(t) = q_0 + q_1 t + \dots + q_Q t^Q$$

]

$$r(t) = p(t) \cdot q(t) = \sum_{n=0}^{P+Q} r_n t^n$$

↑

$$r_n = \sum_{k=\max\{0, n-Q\}}^{\min\{n, P\}} p_k q_{n-k}, \quad 0 \leq n \leq P+Q$$

8

of degree p and q respectively, the product will be a polynomial of degree p plus q and each coefficient in the product can be computed using this formula here. Now the limits in this summation are a bit complicated

notes

summary

5m 13s



Polynomial multiplication and convolution are the same



$$p(t) = p_0 + p_1 t + \dots + p_P t^P$$

$$q(t) = q_0 + q_1 t + \dots + q_Q t^Q$$

$$r(t) = p(t) \cdot q(t) = \sum_{n=0}^{P+Q} r_n t^n$$

$$r_n = \sum_{k=\max\{0, n-Q\}}^{\min\{n, P\}} p_k q_{n-k}, \quad 0 \leq n \leq P+Q$$

but we can simplify this if we just assume that all coefficients in the original polynomials p of t and q of t are equal to 0 when the index is outside of the 0 p range for the first polynomial and the 0 q range for the second polynomial. With this convention we can see that the

notes

summary

5m 27s



Polynomial multiplication and convolution are the same



Better still: assume $p_n = 0$ for $n \notin [0, P]$ and $q_n = 0$ for $n \notin [0, Q]$:

$$r_n = \sum_{k=-\infty}^{\infty} p_k q_{n-k},$$

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



9

coefficient number n of the product polynomial is given by this formula here and if we compare this formula to the formula for the convolution of the sequences we see that the two are exactly the same. Now why is this important? Well it's now time to introduce the Z-transform and its properties. So given an infinite sequence x of n

notes

summary

5m 50s



$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad z \in \mathbb{C}$$

- ▶ associate a power series (i.e. a polynomial) to a sequence
- ▶ for us mostly a formal operator...
- ▶ ...but also as the extension of the DTFT to the whole complex plane:

$$X(z)|_{z=e^{j\omega}} = \text{DTFT}\{x[n]\}$$

- ▶ (and now the notation $X(e^{j\omega})$ should make more sense)

the Z-transform is a complex valued function of a complex variable z defined as the sum for n that goes from minus infinity to plus infinity of x of n times z to the minus n . You can see that we're basically transforming

notes

summary

6m 10s



$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad z \in \mathbb{C}$$

- ▶ associate a power series (i.e. a polynomial) to a sequence
- ▶ for us mostly a *formal operator*...
- ▶ ...but also as the extension of the DTFT to the whole complex plane:

$$X(z)|_{z=e^{j\omega}} = \text{DTFT} \{x[n]\}$$

- ▶ (and now the notation $X(e^{j\omega})$ should make more sense)

a sequence into a polynomial, actually a power series given that the index goes from minus infinity to plus infinity. And for us this is mostly a formal operator but you can see that the formula looks a lot like the formula for the DTFT and as a matter of fact

notes

summary

6m 28s



- $$X(z)|_{z=e^{j\omega}} = \text{DTFT} \{x[n]\}$$

- 10

notes

6m 44s



$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad z \in \mathbb{C}$$

- ▶ associate a power series (i.e. a polynomial) to a sequence
- ▶ for us mostly a *formal operator*...
- ▶ ...but also as the extension of the DTFT to the whole complex plane:

$$X(z)|_{z=e^{j\omega}} = \text{DTFT} \{x[n]\}$$

- ▶ (and now the notation $X(e^{j\omega})$ should make more sense)

to denote the DTFT. There is sort of a continuity

notes

summary

7m 0s



$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad z \in \mathbb{C}$$

- ▶ associate a power series (i.e. a polynomial) to a sequence
- ▶ for us mostly a *formal operator*...
- ▶ ...but also as the extension of the DTFT to the whole complex plane:

$$X(z)|_{z=e^{j\omega}} = \text{DTFT} \{x[n]\}$$

- ▶ (and now the notation $X(e^{j\omega})$ should make more sense)

between the Z-transform and the DTFT. So the Z-transform is a power series and so as such

notes

summary

7m 1s



linearity:

$$\mathcal{Z}\{\alpha x[n] + \beta y[n]\} = \alpha X(z) + \beta Y(z)$$

time shift:

$$\mathcal{Z}\{x[n - N]\} = z^{-N}X(z)$$

convolution:

$$\mathcal{Z}\{h[n] * x[n]\} = H(z)X(z)$$

12

we should be concerned with its convergence. And we will, but not for the moment. For the moment we're more interested in its properties and we will start with the simplest one which is linearity. This is very easy to verify. So if you have a linear combination of two sequences and you take the Z-transform of this linear combination you end up with the linear combination of the Z-transforms. Another very important

notes

summary

7m 13s



linearity:

$$\mathcal{Z}\{\alpha x[n] + \beta y[n]\} = \alpha X(z) + \beta Y(z)$$

time shift:

$$\mathcal{Z}\{x[n - N]\} = z^{-N}X(z)$$

convolution:

$$\mathcal{Z}\{h[n] * x[n]\} = H(z)X(z)$$

property that is

notes

summary

7m 36s



linearity:

$$\mathcal{Z}\{\alpha x[n] + \beta y[n]\} = \alpha X(z) + \beta Y(z)$$

time shift:

$$\mathcal{Z}\{x[n - N]\} = z^{-N}X(z)$$

convolution:

$$\mathcal{Z}\{h[n] * x[n]\} = \underbrace{H(z)X(z)}$$

12

immediate to verify is the time shift property. If you take the Z-transform of a delayed sequence, suppose you take a sequence x of n , you delay it by big N samples, the result is the original Z-transform multiplied by a delayed term which is Z to the power of minus capital N . And finally the most important property of all is the convolution property. If you take the Z-transform of the convolution of two sequences you end up with the product of the Z-transform. And this is just an extension of the polynomial multiplication example that we saw before where we just use finite length polynomials. We multiply these polynomials, the resulting coefficients will be the convolution of the sequences that represent the coefficients. Well here you have the general expression of this property. You take the convolution of two sequences in the Z-domain this convolution is represented by the product of the associated power series. Okay so we took a lot of tangents so we should remember what we're trying to do which is find the frequency response of a linear time invariant system starting from the constant coefficient

notes

summary

7m 38s



$H(z)$ is the *transfer function* of the system

13



Consider an LTI system with impulse response $h[n]$

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

$$\mathcal{Z}\{y[n]\} = \mathcal{Z}\{h[n] * x[n]\}$$

$$Y(z) = H(z)X(z)$$

$H(z)$ is the *transfer function* of the system

13

will be the convolution of the input with the impulse response.

notes

summary

9m 1s



Consider an LTI system with impulse response $h[n]$

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

$$\mathcal{Z}\{y[n]\} = \mathcal{Z}\{h[n] * x[n]\}$$

$$Y(z) = H(z)X(z)$$

$H(z)$ is the *transfer function* of the system

13

If we now move on to the Z-domain we find out that the Z-transform of the output will be the Z-transform of this convolution. Namely we can express the Z-transform of the output

notes

summary

9m 5s



Consider an LTI system with impulse response $h[n]$

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

$$\mathcal{Z}\{y[n]\} = \mathcal{Z}\{h[n] * x[n]\}$$

$$\underline{Y(z)} = \underline{H(z)X(z)}$$

$H(z)$ is the *transfer function* of the system

13

as the product of the Z-transform of the input times the Z-transform of the impulse response. This function H of z is called the transfer function of the system. Now this is not particularly surprising. The transfer function is the Z-transform of the impulse response

notes

summary

9m 17s



- ▶ the transfer function is the z-transform of the impulse response
- ▶ by setting $z = e^{j\omega}$ in $H(z)$ we get the frequency response

and if you set z equal to $e^{j\omega}$ as we said before

notes

summary

9m 36s



- ▶ the transfer function is the z-transform of the impulse response
- ▶ by setting $z = e^{j\omega}$ in $H(z)$ we get the frequency response

$$Y(z) = H(z)X(z)$$

$$z = e^{j\omega}$$

$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

you get the frequency response so if you have Y of z that is equal to H of z X of z and you replace z by $e^{j\omega}$ you have that Y of $e^{j\omega}$ is equal to H of $e^{j\omega}$ times X of $e^{j\omega}$

notes

summary

9m 39s



- ▶ the transfer function is the z-transform of the impulse response
- ▶ by setting $z = e^{j\omega}$ in $H(z)$ we get the frequency response

$$Y(z) = H(z)X(z)$$

$$z = e^{j\omega}$$

$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

14

times X of e to the $j\omega$ and this is just the convolution theorem that we saw before. But the thing is we don't have the impulse response of the system.

notes

summary

9m 57s



$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$$

$$Y(z) \sum_{k=0}^N a_k z^{-k} = X(z) \sum_{k=0}^M b_k z^{-k}$$

We have the constant coefficient difference equation. The algorithmic description of the system. So let's take the constant coefficient difference equation and apply the Z-transform to the left hand side and right hand side. We will exploit the linearity property of the Z-transform

notes

summary

10m 7s



z/ax y

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$$

$$Y(z) \sum_{k=0}^N a_k z^{-k} = X(z) \sum_{k=0}^M b_k z^{-k}$$

and the delay property of the Z-transform so that for instance the Z-transform of a term like $a_k Y$ of n minus k will be equal to a_k times z to the minus k times the Z-transform of Y .

notes

summary

10m 24s



$\mathcal{Z}\{a_k y[n-k]\}$

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$$

$$Y(z) \sum_{k=0}^N a_k z^{-k} = X(z) \sum_{k=0}^M b_k z^{-k}$$

And so if we do this for all terms and we collect the common factors

notes

summary

10m 35s



$$\mathcal{Z}\{a_k y[n-k]\} =$$

$$a_k z^{-k} Y(z)$$

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$$

$$Y(z) \sum_{k=0}^N a_k z^{-k} = X(z) \sum_{k=0}^M b_k z^{-k}$$

we obtain that the relationship between the input and the output is given by this equation here.

notes

summary

10m 37s



$$\mathcal{Z}\{a_k y[n-k]\} =$$

$$a_k z^{-k} Y(z)$$

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$$

$$Y(z) \sum_{k=0}^N a_k z^{-k} = X(z) \sum_{k=0}^M b_k z^{-k}$$



And now the ratio between the input and the output namely the transfer function of the system

notes

summary

10m 49s



$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$$

$$Y(z) \sum_{k=0}^N a_k z^{-k} = X(z) \sum_{k=0}^M b_k z^{-k}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

is given to us directly from the values of the coefficients of the CCDE. As a matter of fact it turns out that the transfer function is a ratio of two finite degree polynomials in Z where the coefficients of these polynomials are given by the coefficients of the constant coefficient difference equation. So we have found an indirect way to obtain the transfer function without having to compute the impulse response explicitly.

notes

summary

10m 54s



- ▶ we can obtain the transfer function of an LTI directly from the CCDE coefficients!
- ▶ the transfer function is a ratio of polynomials
- ▶ this ASSUMES that everything converges...

Again this assumes that all the Z-transforms that we have used liberally do actually exist and converge but we will worry about this in a second. For now let's look a bit more at the general structure of the rational transfer function which is the type of transfer function that you will always get

notes

summary

11m 29s



$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

► feedforward part

► feedback part

when dealing with linear time invariant systems that you can implement. So this is a ratio of two polynomials in Z and the numerator here captures the feed forward part of the system.

notes

summary

11m 44s



$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

► feedforward part
 ► feedback part

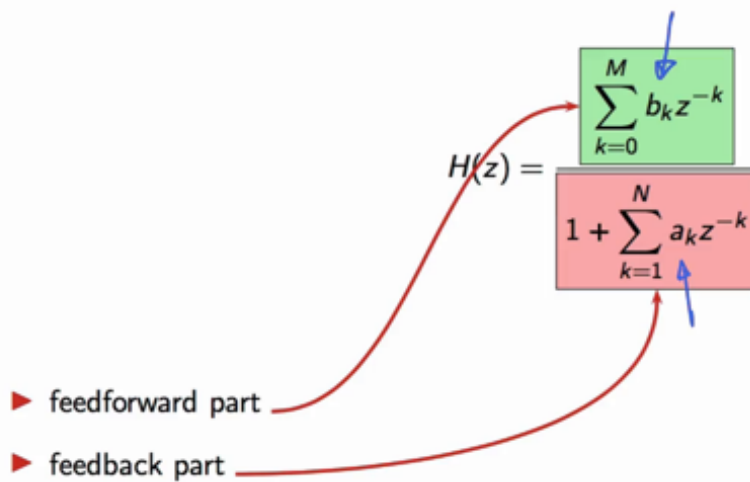
The coefficients here for the numerator are the coefficients in the CCDE

notes

summary

11m 58s





that compute a linear combination of past input samples whereas the denominator captures the feedback part of the system namely the coefficients here are the coefficients that in the CCDE compute a linear combination of output samples. For purely feed forward systems like the moving average

notes

summary

12m 4s



Leaky Integrator revisited



▶ CCDE: $y[n] = (1 - \lambda)x[n] + \lambda y[n - 1]$

▶ impulse response: $h[n] = (1 - \lambda)\lambda^n u[n]$

▶ transfer function from impulse response

$$H(z) = (1 - \lambda) \sum_{n=0}^{\infty} \lambda^n z^{-n} = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

▶ transfer function from CCDE:

$$Y(z) = (1 - \lambda)X(z) + \lambda z^{-1}Y(z)$$

$$H(z) = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

18

the denominator will be equal to 1 because all the a_k coefficients will be equal to 0. Let's consider an example. Remember the Leaky integrator? The Leaky integrator was introduced via its CCDE which is a way to compute a sort of a running average of the current input

notes

summary

12m 25s



Leaky Integrator revisited



► CCDE: $y[n] = \underbrace{(1 - \lambda)x[n]} + \underbrace{\lambda y[n - 1]}$]

► impulse response: $h[n] = (1 - \lambda)\lambda^n u[n]$

► transfer function from impulse response

$$H(z) = (1 - \lambda) \sum_{n=0}^{\infty} \lambda^n z^{-n} = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

► transfer function from CCDE:

$$Y(z) = (1 - \lambda)X(z) + \lambda z^{-1}Y(z)$$

$$H(z) = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

18

by updating the previous estimate of the average with a little portion of the current input.
Since the CCDE for the Leaky integrator is very simple

notes

summary

12m 42s



Leaky Integrator revisited



- ▶ CCDE: $y[n] = (1 - \lambda)x[n] + \lambda y[n - 1]$
- ▶ impulse response: $h[n] = (1 - \lambda)\lambda^n u[n]$
- ▶ transfer function from impulse response

$$H(z) = (1 - \lambda) \sum_{n=0}^{\infty} \lambda^n z^{-n} = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

- ▶ transfer function from CCDE:

$$Y(z) = (1 - \lambda)X(z) + \lambda z^{-1}Y(z)$$

$$H(z) = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

18

we could compute the impulse response by hand and we found this expression here. It's an exponentially decaying sequence multiplied by the step signal.

notes

summary

12m 55s



Leaky Integrator revisited



- ▶ CCDE: $y[n] = (1 - \lambda)x[n] + \lambda y[n - 1]$
- ▶ impulse response: $h[n] = (1 - \lambda)\lambda^n u[n]$
- ▶ transfer function from impulse response

$$H(z) = (1 - \lambda) \sum_{n=0}^{\infty} \lambda^n z^{-n} = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

- ▶ transfer function from CCDE:

$$Y(z) = (1 - \lambda)X(z) + \lambda z^{-1}Y(z)$$

$$H(z) = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

18

We can now compute the transfer function directly from the impulse response because we have it. So we take the Z-transform of the impulse response. This is a simple application of the geometric sum

notes

summary

13m 8s



Leaky Integrator revisited



- ▶ CCDE: $y[n] = (1 - \lambda)x[n] + \lambda y[n - 1]$
- ▶ impulse response: $h[n] = (1 - \lambda)\lambda^n u[n]$
- ▶ transfer function from impulse response

$$H(z) = (1 - \lambda) \underbrace{\sum_{n=0}^{\infty} \lambda^n z^{-n}} = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

- ▶ transfer function from CCDE:

$$Y(z) = (1 - \lambda)X(z) + \lambda z^{-1}Y(z)$$

$$H(z) = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

18

and if we take the Z-transform we attain that the transfer function is 1 minus lambda divided by 1 minus lambda times Z to the minus 1. But now see how easy it is to arrive exactly at the same result without having to compute the impulse response explicitly. We just take the Z-transform of the CCDE directly

notes

summary

13m 20s



Leaky Integrator revisited



- ▶ CCDE: $y[n] = (1 - \lambda)x[n] + \lambda y[n - 1]$
- ▶ impulse response: $h[n] = (1 - \lambda)\lambda^n u[n]$
- ▶ transfer function from impulse response

$$H(z) = (1 - \lambda) \sum_{n=0}^{\infty} \lambda^n z^{-n} = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

- ▶ transfer function from CCDE:

$$Y(z) = (1 - \lambda)X(z) + \lambda z^{-1}Y(z)$$

$$H(z) = \frac{(1 - \lambda)}{1 - \lambda z^{-1}}$$

18

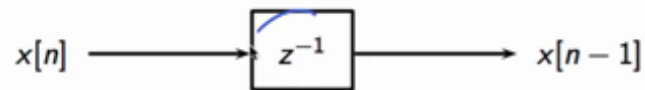
and we have Y of Z equal to 1 minus lambda times X of Z plus lambda using the delay property Z minus 1 times Y of Z. And then if we collect Y of Z we obtain the ratio of Y of Z over X of Z as 1 minus lambda divided by 1 minus lambda Z to the minus 1. As a final note now that we're familiar with the Z-transform formalism

notes

summary

13m 39s





$$Y(z) = z^{-1} X(z)$$

now the notation should make more sense!

we can understand why the delay block in our block diagrams were labeled with Z to the minus 1. Because Z to the minus 1 is the transfer function of an LTI system

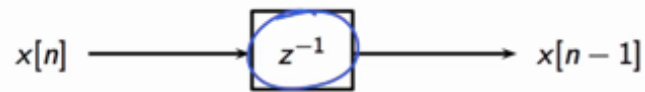
notes

summary

14m 5s



BTW, remember the delay block?



$$Y(z) = z^{-1} X(z)$$

now the notation should make more sense!

19

whose sole action is to delay the input by 1.

notes

summary

14m 15s

