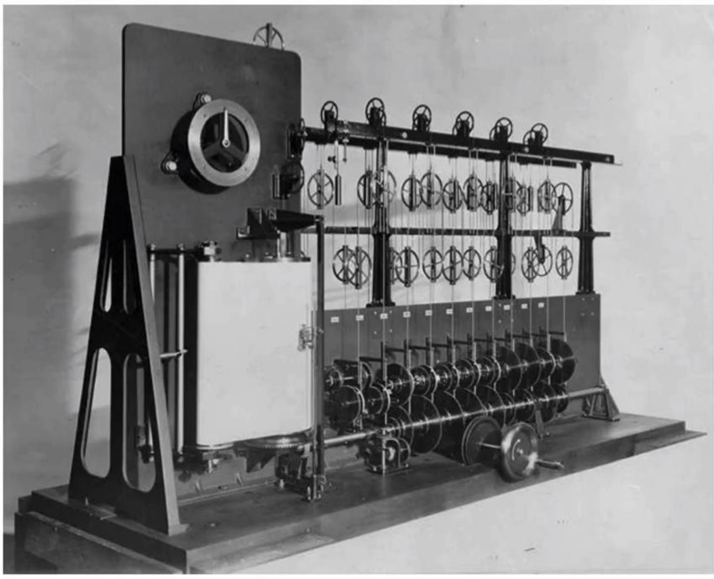
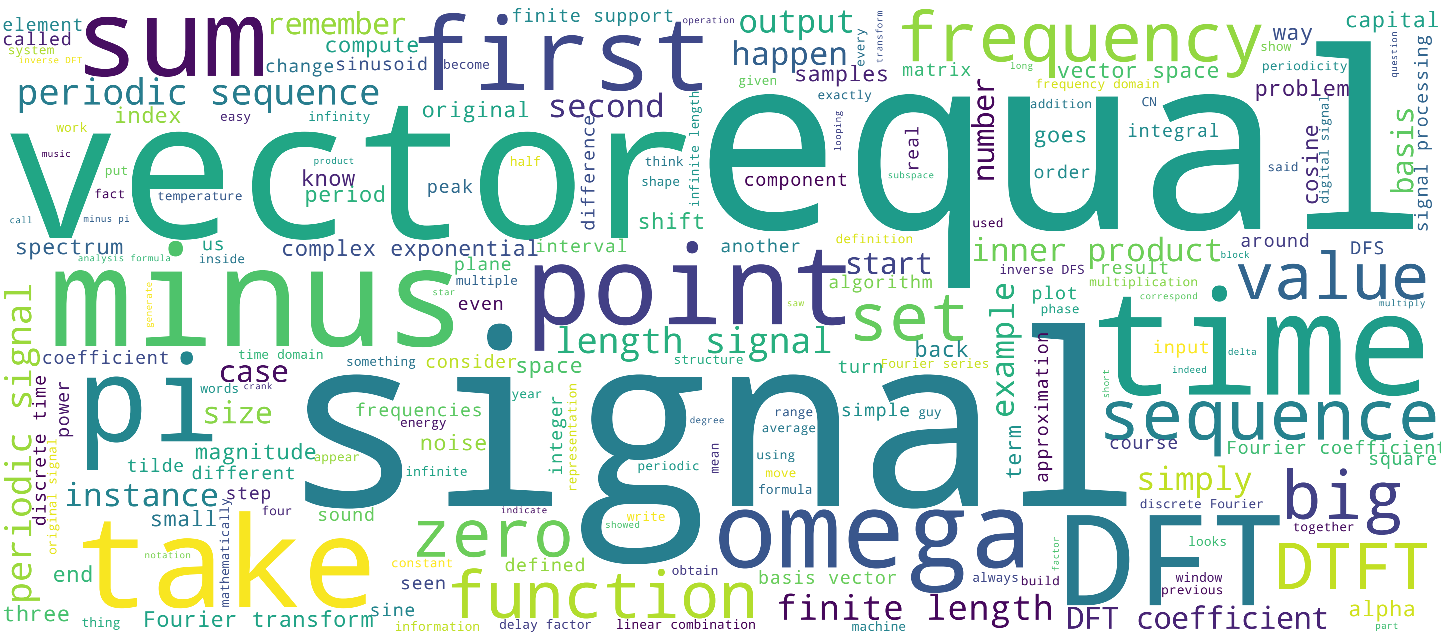
The machine before DSP



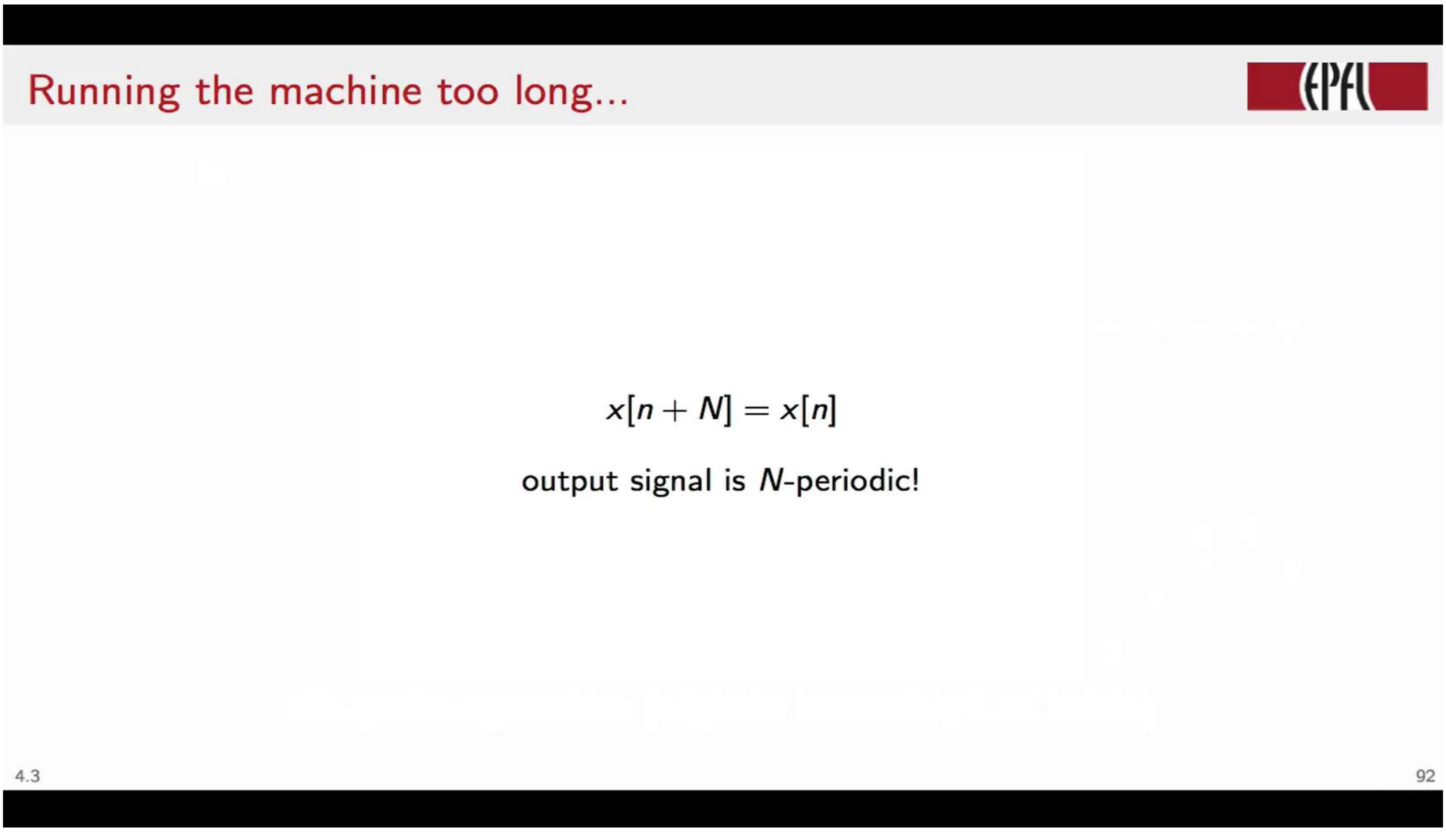
tide-predicting machine (originally invented by Lord Kelvin)



Search MOOC

Video

EPFL



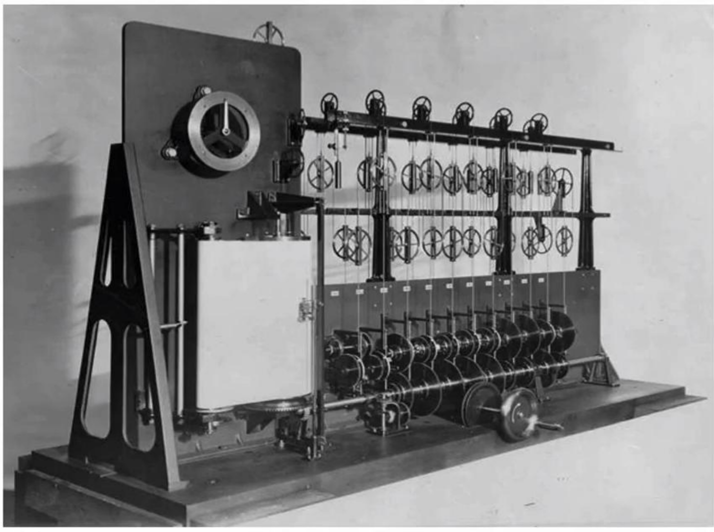
Remember a few lectures ago we talked about Lord Kelvin's tide-prediction machine. This was a mechanical device that was used to predict the tide based on a set of Fourier coefficients. So it implemented mechanically the DFT reconstruction formula.

Notes

Summary

0m 01s

The machine before DSP



tide-predicting machine (originally invented by Lord Kelvin)

4.3 91

You would set the dials to the magnitude and phase of each Fourier coefficient and then turn in the crank, the machine would generate a curve that was the inverse DFT of the set of coefficients that you set up in the machine. So now here is a question.

Notes

Summary

0m 18s

Running the machine too long...

$$x[n+N] = x[n]$$

output signal is N-periodic!

4.3

92

What happens if you turn the crank too long so that you go over the range of indices that are allowed for by the standard DFT formula? Well it turns out that the output will become periodic in other words, X of small N plus big N will be equal to X of N.

Notes

Summary

0m 31s

Inherent periodicities in the DFT

the synthesis formula:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}nk}, \qquad n = 0, 1, \ldots, N-1$$

produces an N-point signal in the time domain

the analysis formula:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}nk}, \qquad k = 0, 1, \ldots, N-1$$

produces N-point signal in the frequency domain

4.3

93

This is actually quite apparent from the structure of the synthesis and analysis formulas of the DFT. Let's start from the synthesis formula. So in theory the output index small n should go from 0 to big N minus 1. But you can see that it only appears here inside this complex exponential. Now this guy is 2 Pi periodic so if you push n over big N minus 1 this will simply cycle over once again as if n was looping over the 0 to N minus 1 range.

Notes

Summary

0m 49s

Inherent periodicities in the DFT

the synthesis formula:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}nk}, \qquad n \in \mathbb{Z}$$

produces an N-periodic signal in the time domain

the analysis formula:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}nk}, \qquad k = 0, 1, \ldots, N-1$$

produces N-point signal in the frequency domain

4.3

93

So in the end you can actually safely take n to be from the set of integers and the output will be an n periodic signal in the time domain. Actually the same holds for the analysis formula. You can let the index K roam over the entire set of integers and since it appears only in this complex exponential here, again it will be as if K was looping over the 0 to N minus 1 range.

Notes

Summary

1m 19s

Inherent periodicities in the DFT

the synthesis formula:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}nk}, \qquad n \in \mathbb{Z}$$

produces an *N*-periodic signal in the time domain

the analysis formula:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}nk}, \qquad k \in \mathbb{Z}$$

produces *N*-periodic signal in the frequency domain

4.3 93

So what happens really is that you can consider the sequence of DFT coefficients as an n periodic signal in the frequency domain.

Notes

Summary

1m 47s

Discrete Fourier Series (DFS)

DFS = DFT with periodicity explicit

- the DFS maps an N-periodic signal onto an N-periodic sequence of Fourier coefficients
- the inverse DFS maps an N-periodic sequence of Fourier coefficients a set onto an N-periodic signal
- the DFS of an N-periodic signal is mathematically equivalent to the DFT of one period

Now when we talk explicitly about the periodicity of the Fourier transform we prefer to use the term discrete Fourier series or DFS for short. So the DFS is just the DFT with the periodicity explicit. The DFS maps an n periodic signal onto an n periodic sequence of Fourier coefficients. And the inverse DFS maps an n periodic sequence of Fourier coefficients onto an n periodic signal. But mathematically the two are the same. It's just the conceptual difference.

Notes

Summary

1m 56s

Finite-length time shifts revisited

The DFS helps us understand how to define time shifts for finite-length signals.

For an N- periodic sequence $\tilde{x}[n]$:

- $\tilde{x}[n-M]$ is well-defined for all $M \in \mathbb{N}$
- $\text{DFS}\{\tilde{x}[n-M]\} = e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k]$ (easy derivation)
- $\text{IDFS}\left\{\tilde{X}[k]\right\} = \tilde{x}[n-M]$

$\tilde{X}[k] = \text{DFS}[\tilde{x}[n]]$

However DFS helps us understand for instance why we said that the natural shift for finite length sequences is actually a periodic shift. Let's consider an n periodic sequence x tilde of n. For periodic sequences shifts are well defined. We can take x tilde of n minus big M and it's just a normal shift of an infinite two sided sequence. If we take the DFS of this periodic sequence with a shift we can work out very easily that the DFT coefficient for index K is equal to the DFT coefficient for index K of the original sequence without the shift. So X of K is simply the DFS of tilde X of N with a phase shift term e to the minus J 2 pi over n big M times k.

Notes

Summary

2m 28s

Finite-length time shifts revisited

The DFS helps us understand how to define time shifts for finite-length signals.

For an N- periodic sequence $\tilde{x}[n]$:

- $\tilde{x}[n-M]$ is well-defined for all $M \in \mathbb{N}$
- $\text{DFS}\{\tilde{x}[n-M]\} = e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k]$ (easy derivation)
- $\text{IDFS}\left\{\boxed{e^{-j\frac{2\pi}{N}Mk}}\,\tilde{X}[k]\right\} = \tilde{x}[n-M]$

delay factor

Similarly it's very easy to verify that the inverse discrete Fourier series of a vector of DFS coefficients multiplied by a phase delay factor is simply the periodic sequence delayed by m. So this term e to the minus j 2 Pi over n Mk is a delay factor in the frequency domain.

Notes

Summary

3m 22s

Finite-length time shifts revisited

For an N-point signal $x[n]$:

- $x[n-M]$ is *not* well-defined
- build $\tilde{x}[n] = x[n \mod N] \Rightarrow \tilde{X}[k] = X[k]$
- IDFT $\left\{ e^{-j\frac{2\pi}{N}Mk} X[k] \right\} =$ IDFS $\left\{ e^{-j\frac{2\pi}{N}Mk} \tilde{X}[k] \right\} = \tilde{x}[n-M] = x[(n-M) \mod N]$
- shifts for finite-length signals are "naturally" circular

So what happens for finite length signals? We saw that in this case a shift is not a well defined operation and we need to embed the signal into an infinite length signal either by building a finite support or a periodic signal. Now we can always build x tilde of n as x of n modules big N. Now mathematically the DFS of this periodic signal is equal to the DFT of the original signal. So the inverse DFT of DFT of the signal times this delay factor that we showed in the previous slide is numerically equal to the inverse DFS of the same quantity. But we know that the inverse DFS of this will be a delayed periodic sequence. And if we go back to the definition of how we build this periodic sequence this will be simply x of n minus big N modules big N. What we have shown here is that if we think of the Fourier representation of a finite length signal, circular shifts are the natural extension of the shift operator to finite length signals because the underlying Fourier representation is the same as that we would use for a periodic signal built on the finite length signal.

Notes

Summary

3m 42s