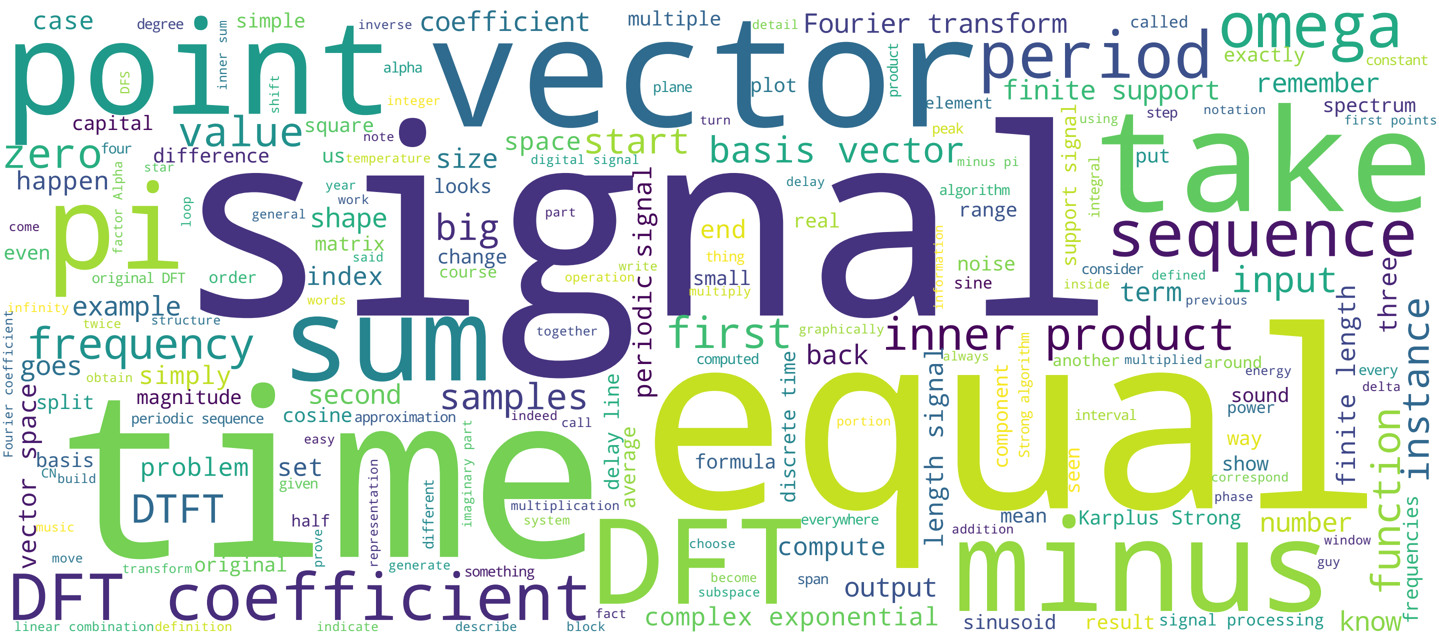
Periodic sequences: a bridge to infinite-length signals

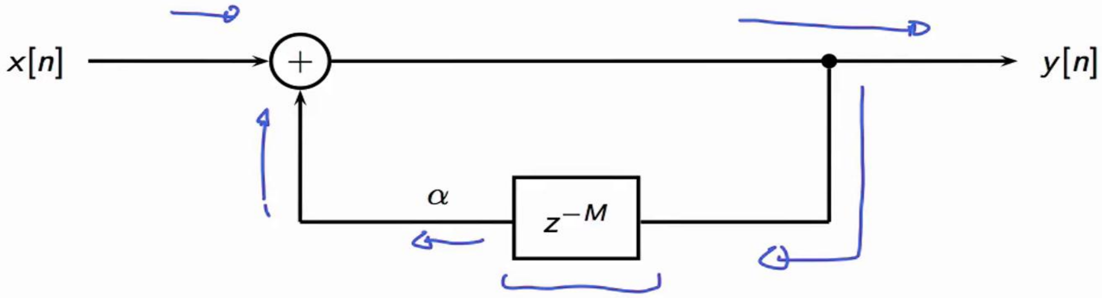
- N-periodic sequence: N degrees of freedom
- DFS: only N Fourier coefficients capture all the information



Search MOOC

Video

Karplus-Strong revisited



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

Let's start with periodic sequences. An N periodic sequence has only N degrees of freedom, and DFS captures the situation by providing us with the spectral representation for the sequence that only has N distinct Fourier coefficients. We can illustrate these points more in detail with the help of the Karplus-Strong algorithm. You remember how the Karplus-Strong circuit works. You have an input, you have a delay line of length M, and an attenuation factor, Alpha. As the samples go in the circuit, they get pushed out to the output, and they get pushed also to the delay line. They will store here an output at the other end of the delay line after M samples, where they get attenuated by a factor, Alpha, and summed to the current input. If we were to describe this mathematically, we would have what we call a difference equation that relates the current output to the past output that has gone through the delay line, so an output delay by M, scaled by the attenuation factor, and summed to the original input.

Notes

Summary

0m 00s

Karplus-Strong revisited

- choose a signal $\bar{x}[n]$ that is nonzero only for $0 \leq n < M$
- $\alpha = 1$ (for now)

$$y[n] = \underbrace{\bar{x}[0], \bar{x}[1], \ldots, \bar{x}[M-1]}_{1^{\text{st}} \text{ period}}, \underbrace{\bar{x}[0], \bar{x}[1], \ldots, \bar{x}[M-1]}_{2^{\text{nd}} \text{ period}}, \underbrace{\bar{x}[0], \bar{x}[1], \ldots}_{\ldots}$$

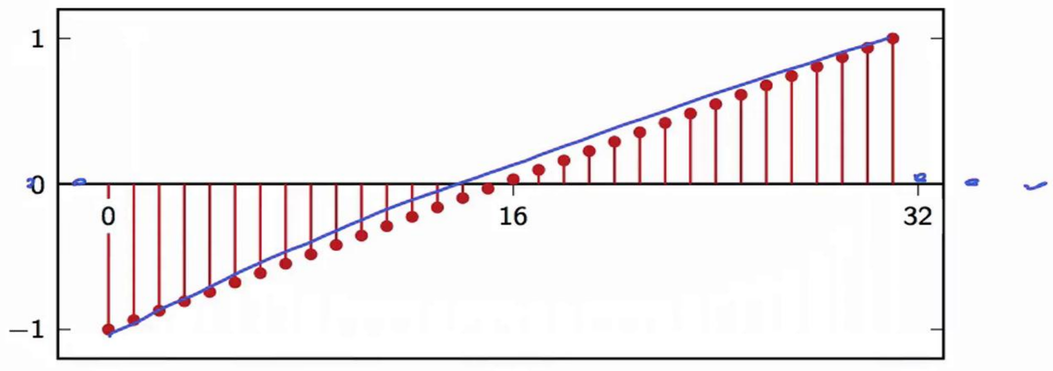
We can use this circuit to generate a simple periodic signal, and to do that, we choose an input signal x that is nonzero, only for n between 0 and M. That means that x of n is actually a finite support sequence, and that's why we use the overbar notation. It will only have M nonzero samples, and it will be 0 everywhere else. Then we choose Alpha equal to 1, so there will be no attenuation in the loop. If we input this signal to the Karplus-Strong circuit, we will see that the output consists of a periodic signal where each period is constituted of the M nonzero samples of the input finite support signal.

Notes

Summary

1m 05s

Example: 32-tap sawtooth wave



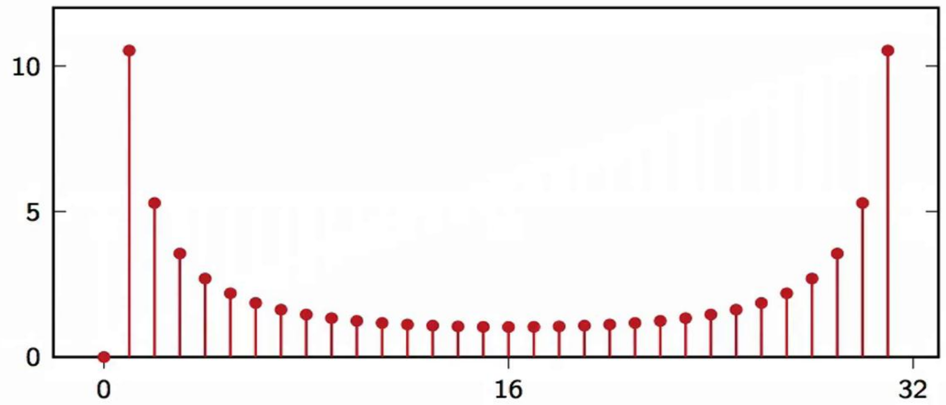
Suppose for instance that our finite support signal looks like this, has a nonzero support of 32 points, and over the support, it looks like a straight line, and it's 0 everywhere else. If we use this as an input to the Karplus-Strong algorithm, we will get a periodic repetition of this shape, which overall will look like a sawtooth wave. If we take the DFT of this signal, considered as a 32-point finite length signal, the DFT will look like this.

Notes

Summary

1m 46s

Example: DFT of 32-tap sawtooth wave



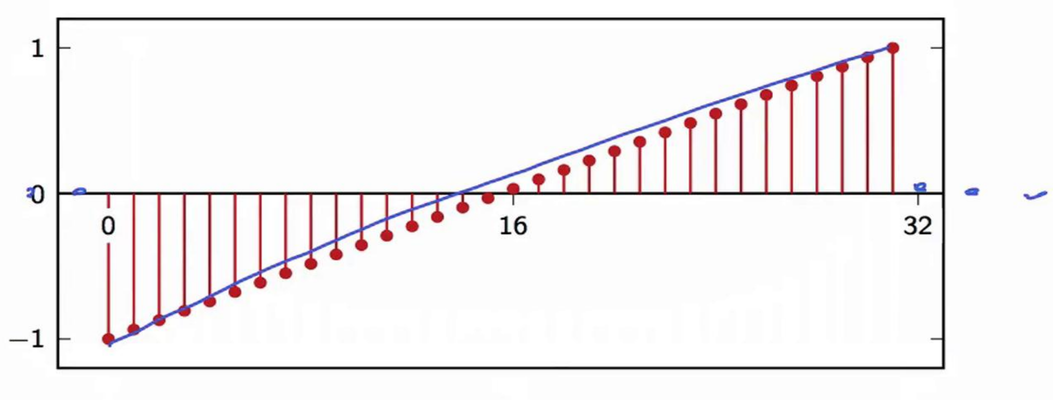
Computing the exact value of the DFT coefficients is left as an exercise. Just note for instance that the coefficient in 0 is 0, because the signal is actually centred around 0, so its average is 0.

Notes

Summary

2m 18s

Example: 32-tap sawtooth wave

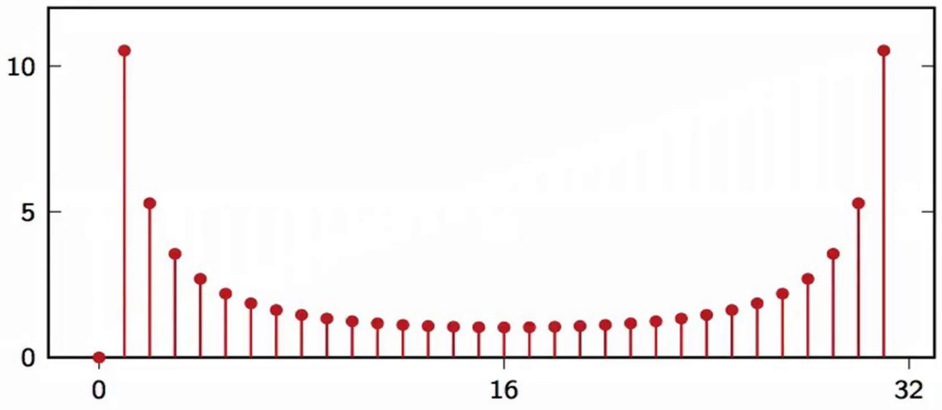


Notes

Summary

2m 28s

Example: DFT of 32-tap sawtooth wave

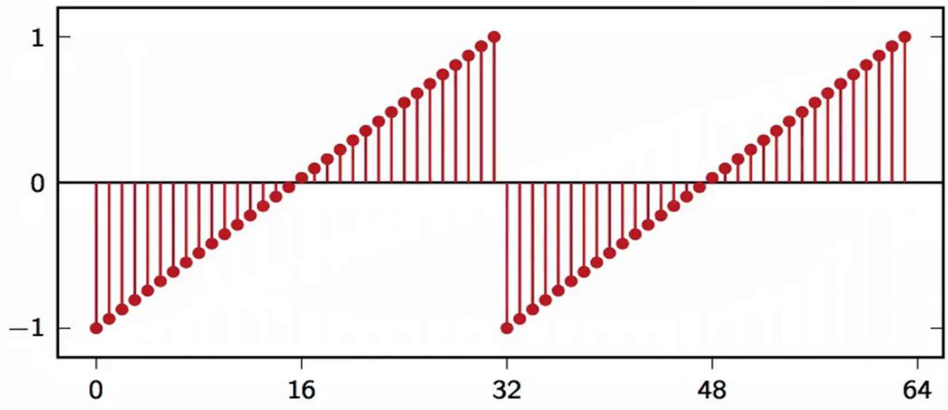


What happens then if we take the DFT of two periods?

Notes

Summary

What if we take the DFT of two periods?



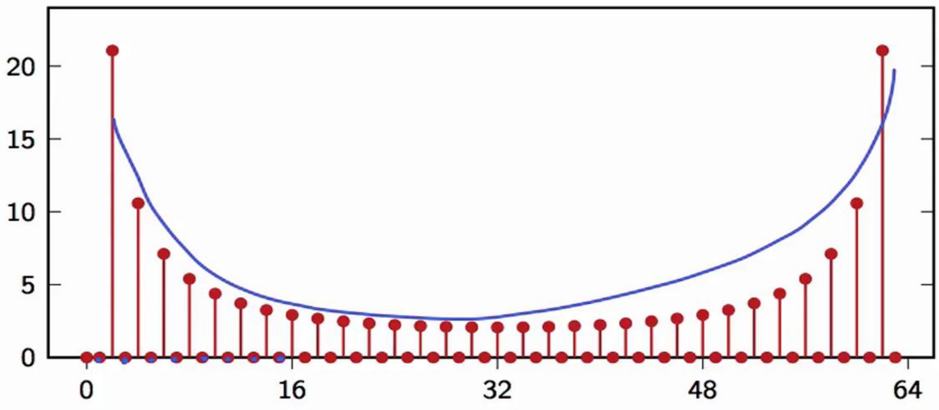
We consider the first 64 points that come out of the Karplus-Strong algorithm, and we take a DFT of this.

Notes

Summary

2m 36s

Example: 64-point DFT of two periods

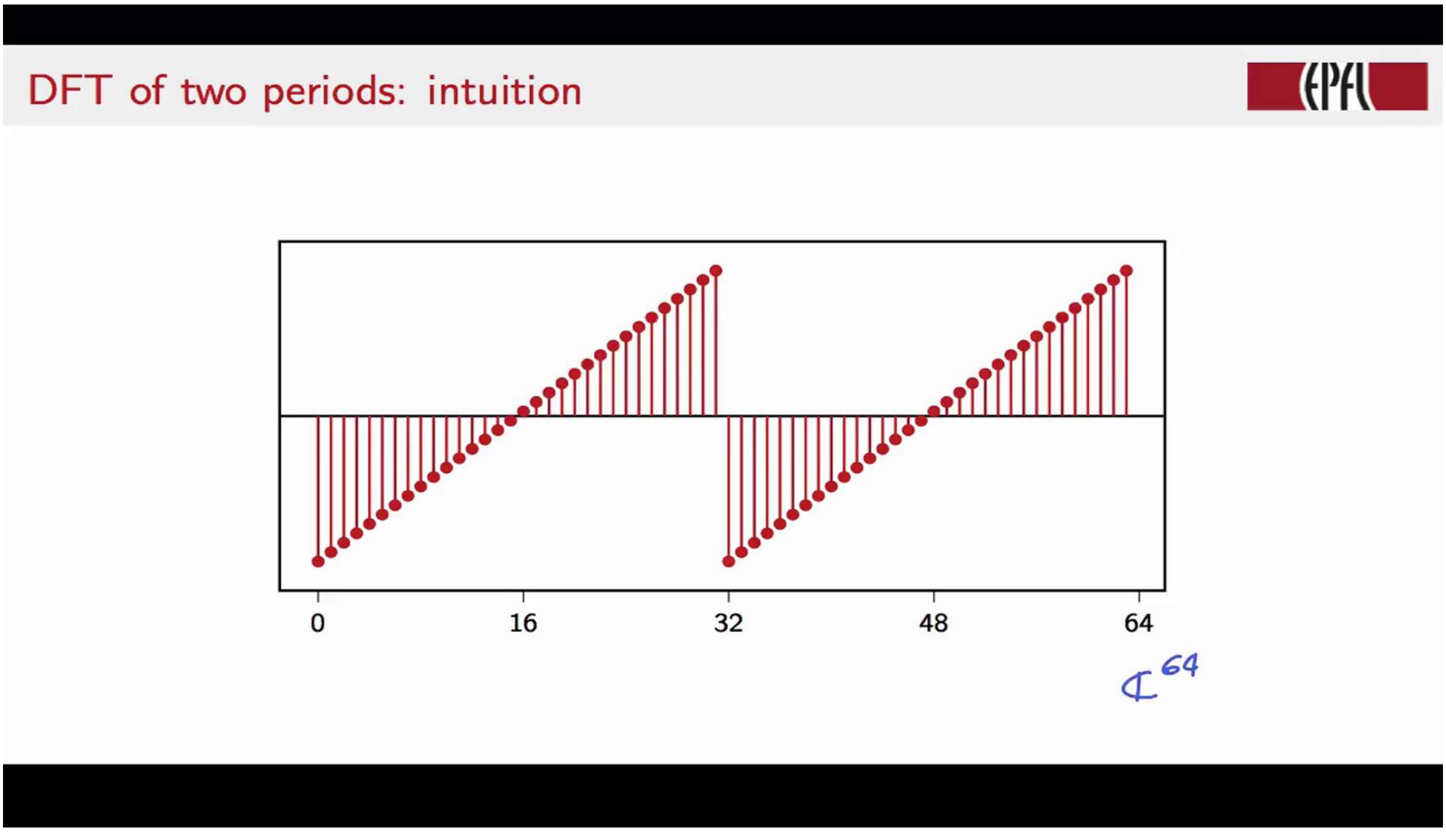


Well, if we go through the math, or even if we put this into a numerical package, we will see that the shape of the DFT coefficient, the envelope, so to speak, is the same as the one for just the first 32 points. However, for every DFT coefficient, there is an extra DFT coefficient with value 0. In other words, the DFT coefficients have been interleaved with zero-valued coefficients. To get an intuition as to why it is so, let's go back to the definition of DFT.

Notes

Summary

2m 44s



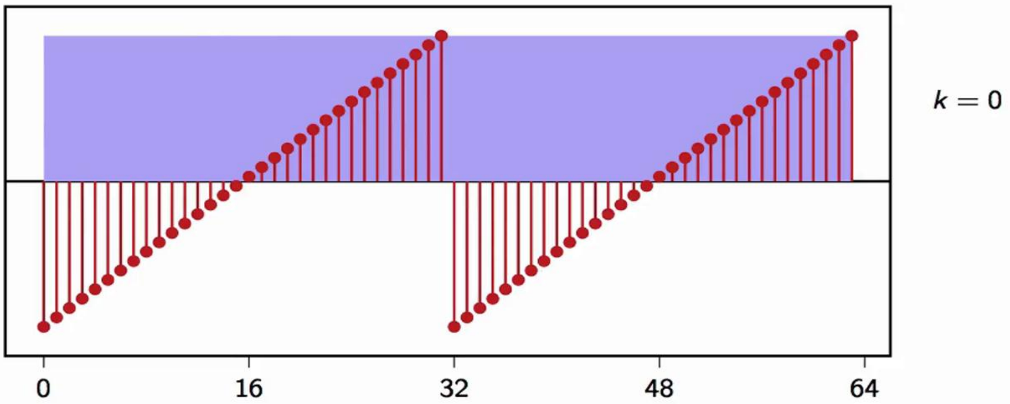
Each DFT coefficient will be the inner product between the appropriate basis vector in the Fourier basis for C^{64}, and this signal that we see here. Let's see graphically how these inner products are computed. Let's start with the first basis vector.

Notes

Summary

3m 18s

DFT of two periods: intuition



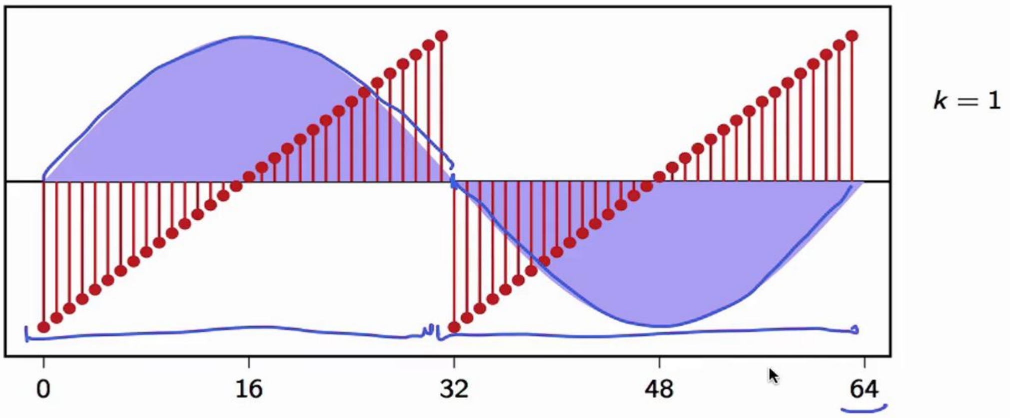
We will show here in blue just the ideal shape of this basis vector. We don't really care about the real and imaginary part because we're just interested in the structure. The first vector is just the constant 1, and the result of the inner product will be the non normalized average of this signal in red. We know the average to be 0. The same holds for two periods, so the first coefficient is 0.

Notes

Summary

3m 33s

DFT of two periods: intuition



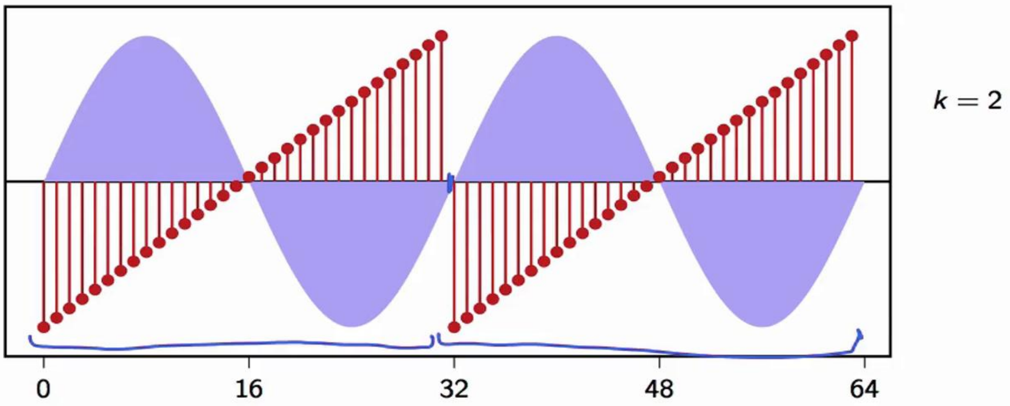
For k equal to 1, we have the first basis vector, and this will be a sinusoid that will span a full period over 64 points. What that means is that the first 32 points will be multiplied by a shape that is exactly the inverse of the shape for the next 32 points. This sub sum of the inner product plus this will give 0.

Notes

Summary

3m 57s

DFT of two periods: intuition



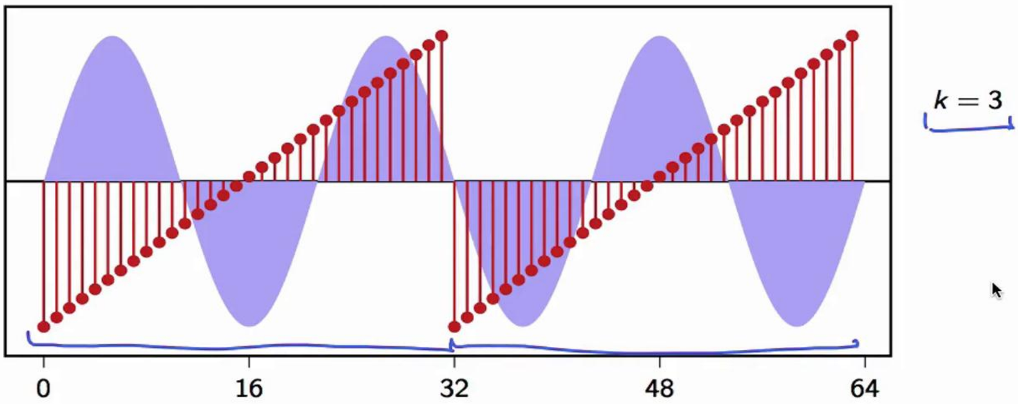
For k equal to 2, on the other hand, the Fourier vector will span two periods over the space of 64 points, and so here we simply have twice the same thing. This half of the inner product will be equal to the second half, and the DFT coefficient will be twice the original DFT coefficient.

Notes

Summary

4m 24s

DFT of two periods: intuition

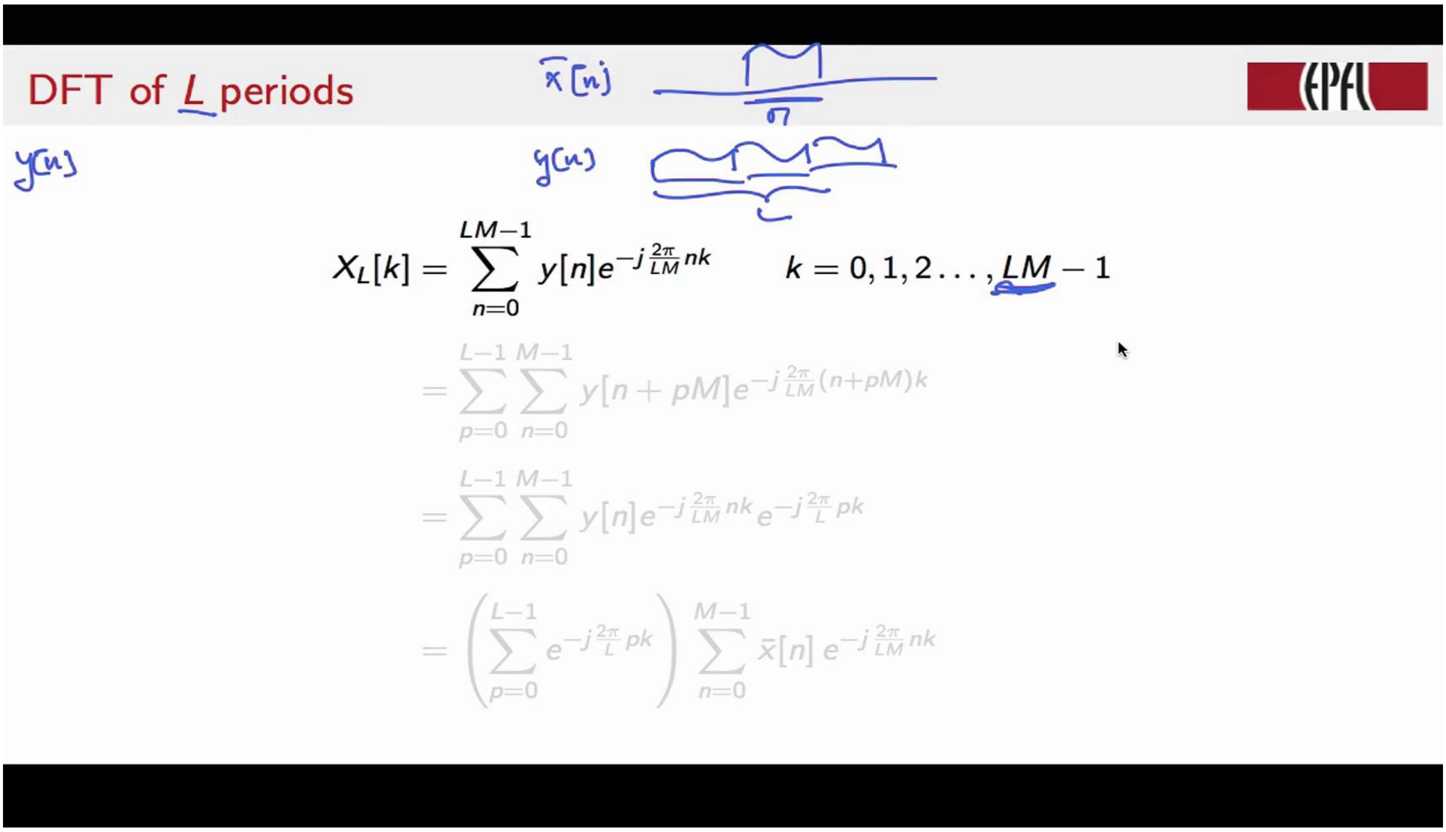


For k equal to 3, we have another situation of anti-symmetry between the first part of the Fourier vector and the second part, and so these two parts will cancel each other out. We can see that in general, all odd index coefficients will be 0 for this two-period signal.

Notes

Summary

4m 44s



We can prove this for an arbitrary number of periods. Suppose y of n is a finite length signal that is composed of L repetitions of the same M points. We said that overbar x of n is a finite support signal that is 0 everywhere except in a portion here. We take this portion, this is M points, and we build y of n by putting L of these pieces together. So this would be L of n. The Fourier transform of y of n is simply given by the sum for n that goes from 0 to LM minus 1, LM being the length of this signal now, of y of n times e to the minus J 2 Pi over LM times n times K. K ranges from 0 to LM minus 1. This is standard stuff. The only difference is that the length of our signal is LM.

Notes

Summary

5m 00s

DFT of L periods

$$X_L[k] = \sum_{n=0}^{LM-1} y[n] e^{-j\frac{2\pi}{LM}nk} \qquad k = 0, 1, 2 \ldots, LM-1$$

$k = n + pM$

$$= \underbrace{\sum_{p=0}^{L-1}}\underbrace{\sum_{n=0}^{M-1}} y[n+pM] e^{-j\frac{2\pi}{LM}\underline{(n+pM)}k}$$

$$= \sum_{p=0}^{L-1}\sum_{n=0}^{M-1} y[n] e^{-j\frac{2\pi}{LM}nk} e^{-j\frac{2\pi}{L}pk}$$

$$= \left(\sum_{p=0}^{L-1} e^{-j\frac{2\pi}{L}pk}\right) \sum_{n=0}^{M-1} \bar{x}[n]\, e^{-j\frac{2\pi}{LM}nk}$$

We can split the sum in the following way. We have an outside sum where we span the number of periods and an inner sum where we go through the points in the period. Here, we simply replace k with n plus pM. So k was ranging over the entire signal, now we split this into period plus index inside the period. We can do the same inside the complex exponential.

Notes

Summary

6m 00s

DFT of L periods

$$X_L[k] = \sum_{n=0}^{LM-1} y[n] e^{-j\frac{2\pi}{LM}nk} \qquad k = 0, 1, 2 \ldots, LM-1$$

$$= \sum_{p=0}^{L-1} \sum_{n=0}^{M-1} y[n+pM] e^{-j\frac{2\pi}{LM}(n+pM)k}$$

$$= \sum_{p=0}^{L-1} \sum_{n=0}^{M-1} y[n] e^{-j\frac{2\pi}{LM}nk} e^{-j\frac{2\pi}{L}pk}$$

$$= \underbrace{\left(\sum_{p=0}^{L-1} e^{-j\frac{2\pi}{L}pk} \right)}_{} \underbrace{\sum_{n=0}^{M-1} \bar{x}[n]\, e^{-j\frac{2\pi}{LM}nk}}_{}$$

Then we can split the complex exponential, and we see that the inner sum is this one here, and this term depends only on p. We can bring it forward and have a product of two sums where we have this term here that ranges over the p index and this term here that ranges over the n index, and the two are independent. But now, this guy here we've seen before.

Notes

Summary

6m 26s

We've seen this before

$$\underbrace{\sum_{p=0}^{L-1} e^{-j\frac{2\pi}{L}pk}} = \begin{cases} \underline{L} & \text{if } k \text{ multiple of } L \\ 0 & \text{otherwise} \end{cases}$$

(remember the orthogonality proof for the DFT basis)

This is exactly the same formula that we use to prove the orthogonality of the DFT basis. We know that this sum will be equal to L if k is a multiple of L, or 0 otherwise.

Notes

Summary

6m 53s

DFT of L periods

$$X_L[k] = \begin{cases} L\,\bar{X}[k/L] & \text{if } k = 0, L, 2L, 3L, \ldots \\ 0 & \text{otherwise} \end{cases}$$

Wth this knowledge, we can plug this back into the formula for the Fourier transform of L periods and find out that the kth DFT coefficient will be equal to L times the original DFT coefficient that you could have computed for the finite support signal if k is a multiple of L, and 0 otherwise. Given the DFT of L periods, you will have only one nonzero DFT coefficient out of L, and this will be equal to L times the original DFT coefficient.

Notes

Summary

7m 08s

DFT and DFS

- again, all the spectral information for a periodic signal is contained in the DFT coefficients of a single period
- to stress the periodicity of the underlying signal, we use the term DFS

Well this was a long way to show that once again all the spectral information for a periodic signal is contained in the DFT of a single period. That's why for periodic signal in the end, we use the DFS.

Notes

Summary

7m 42s