

Course material

Course:

Digital Signal Processing II

Video:

2.2.1.c Frequency sampling

Concepts (extracted from automatically generated subtitles):

Frequency sampling. Desired frequency response. Inverse dft. Ideal filter. Frequency response. Frequency domain. New method. Finite set of frequency points. Samples of this frequency characteristic. Fourier transform. Indicator function. Dft of the finite support filter. Classic ideal low pass. Terms of the fine control. Good methods.



[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/>

Frequency sampling

Idea #2:

- ▶ draw desired frequency response $H(e^{j\omega})$
- ▶ take M values at $\omega_k = (2\pi/M)k$
- ▶ compute IDFT of values
- ▶ use result as an M -tap impulse response $\hat{h}[n]$

Let's move on now to a new method to approximate an ideal filter called frequency sampling.

notes

summary

0m 0s



Frequency sampling

Idea #2:

- ▶ draw desired frequency response $H(e^{j\omega})$
- ▶ take M values at $\omega_k = (2\pi/M)k$
- ▶ compute IDFT of values
- ▶ use result as an M -tap impulse response $\hat{h}[n]$

The idea here is to say what if we draw the desired frequency response in the frequency domain, then what we do is we sample this frequency response at regularly spaced intervals,

notes

summary

0m 1s



Frequency sampling

Idea #2:

- ▶ draw desired frequency response $H(e^{j\omega})$
- ▶ take M values at $\omega_k = (2\pi/M)k$
- ▶ compute IDFT of values
- ▶ use result as an M -tap impulse response $\hat{h}[n]$

and then we compute the IDFT of this value, so the inverse DFT, which we can always do for a finite set of frequency points,

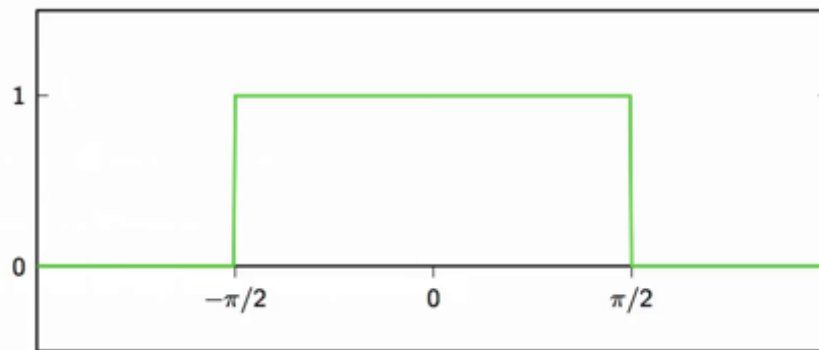
notes

summary

0m 13s



Frequency sampling: desired response



and use the result as an m-tab impulse response $\hat{h}$ of n . Look at an example, we start with our classic ideal low pass with ω_c equal to π over 2 ,

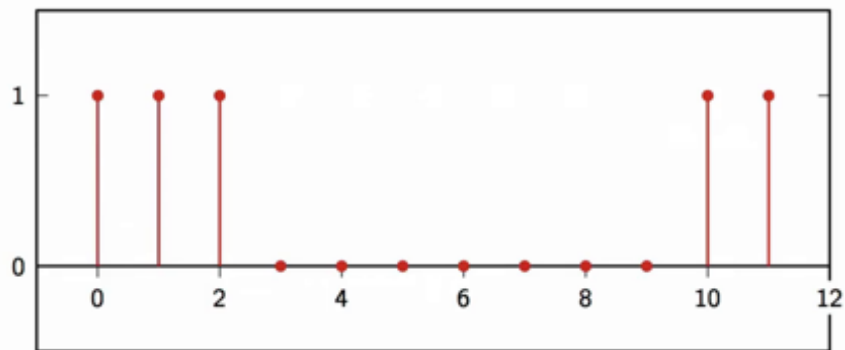
notes

summary

0m 25s



Frequency sampling: DFT samples



and say we pick m equal to 11, and we take 11 samples of this frequency characteristic between minus π and π . We rearrange them according to the DFT notation,

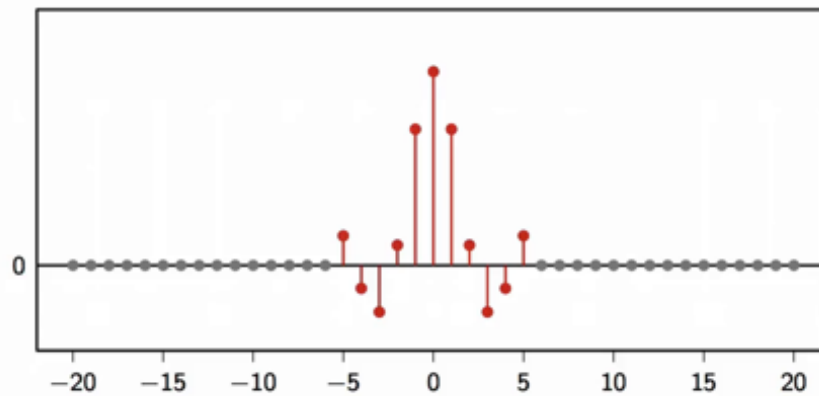
notes

summary

0m 37s



Frequency sampling: impulse response from IDFT



and we compute the inverse DFT with a numeric algorithm. So if we do that, we get the following impulse response. This is an 11-point impulse response obtained by computing the inverse DFT

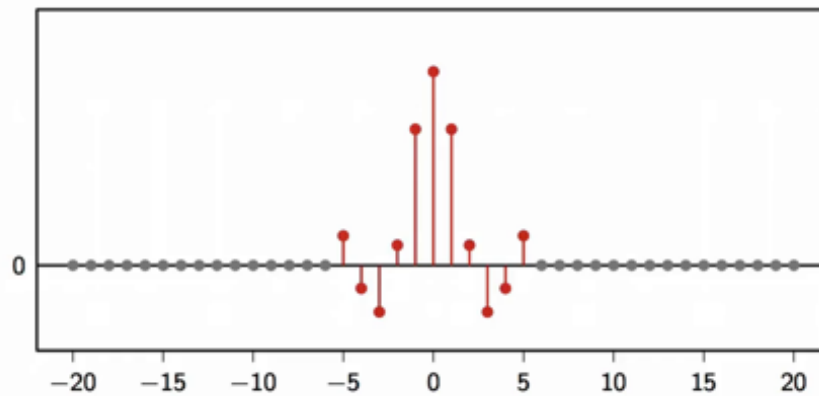
notes

summary

0m 49s



Frequency sampling: impulse response from IDFT



of the samples of the ideal frequency characteristic. So if we take the Fourier transform now, so the DTFT of the finite support filter,

notes

summary

1m 1s



Why it's not such a good idea

- ▶ frequency response is DTFT of finite-support, whose DFT we know
- ▶ frequency response is interpolation of frequency samples
- ▶ interpolator is transform of N -tap rectangular window
- ▶ again, no control over mainlobe and sidelobes

well, what do we get? We need to convert the DFT representation to the DTFT representation.

notes

summary

1m 9s



Why it's not such a good idea

- ▶ frequency response is DTFT of finite-support, whose DFT we know
- ▶ frequency response is interpolation of frequency samples
- ▶ interpolator is transform of N -tap rectangular window
- ▶ again, no control over mainlobe and sidelobes

We know how to do that from the lecture about the relationships between transforms.

notes

summary

1m 13s



Why it's not such a good idea

- ▶ frequency response is DTFT of finite-support, whose DFT we know
- ▶ frequency response is interpolation of frequency samples
- ▶ interpolator is transform of N -tap rectangular window
- ▶ again, no control over mainlobe and sidelobes

And we know that the frequency response will be the interpolation in frequency of the frequency samples that we took in the first step. The interpolator turns out to be, once again, the transform of an n -tap rectangular window.

notes

summary

1m 25s



Why it's not such a good idea

- ▶ frequency response is DTFT of finite-support, whose DFT we know
- ▶ frequency response is interpolation of frequency samples
- ▶ interpolator is transform of N -tap rectangular window
- ▶ again, no control over mainlobe and sidelobes

So we're not really escaping from the indicator function that we used in the impulse truncation method. And because of that, we will have no control over main lobes and side lobes of the

notes

summary

1m 37s



Why it's not such a good idea

- ▶ frequency response is DTFT of finite-support, whose DFT we know
- ▶ frequency response is interpolation of frequency samples
- ▶ interpolator is transform of N -tap rectangular window
- ▶ again, no control over mainlobe and sidelobes

interpolator. Graphically, it looks like this. Here is the original DTFT in green. In gray,

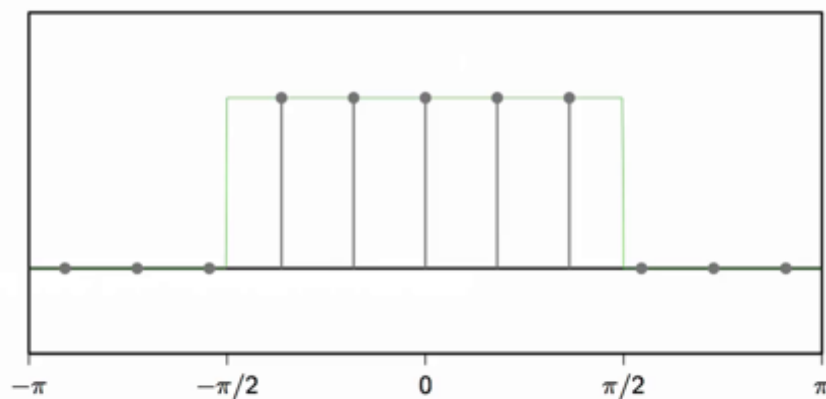
notes

summary

1m 49s



Frequency sampling: frequency response



we have the frequency samples. And the resulting DTFT will be the interpolation of these frequency

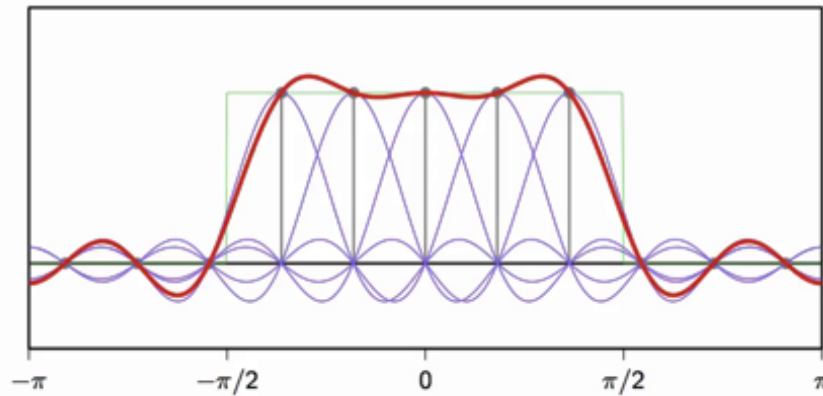
notes

summary

1m 53s



Frequency sampling: frequency response



samples using this interpolator here in blue. When we do the interpolation, that is what we

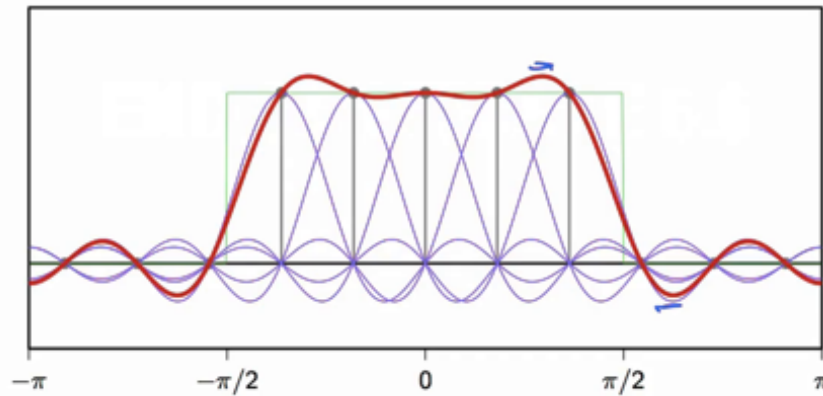
notes

summary

2m 1s



Frequency sampling: frequency response



obtain in the end. And again, we have a filter that will exhibit an error around the passband that we are not going to be able to control. In short, these methods to approximate ideal filters

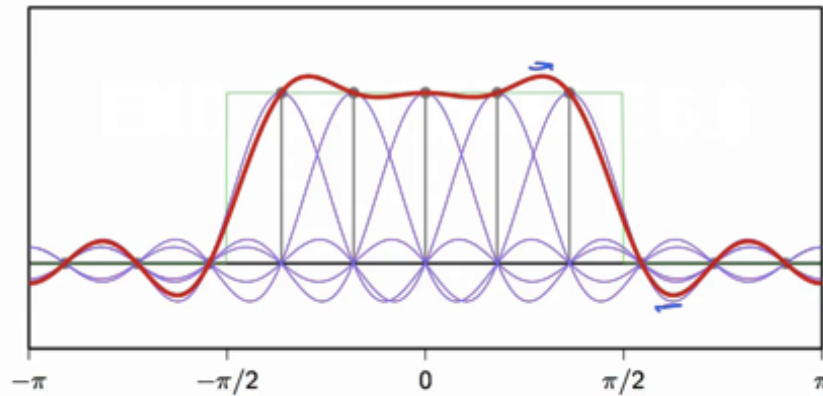
notes

summary

2m 13s



Frequency sampling: frequency response



are certainly very useful when we want to derive a quick and dirty prototype.

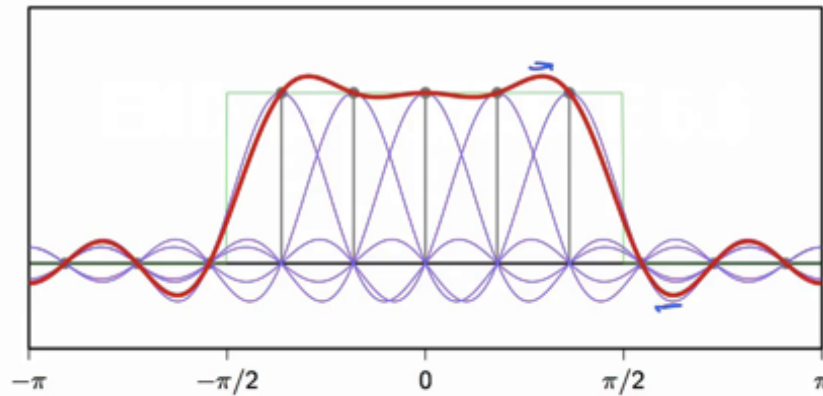
notes

summary

2m 24s



Frequency sampling: frequency response



And we don't have time to use more sophisticated filter design methods that we will see later on.

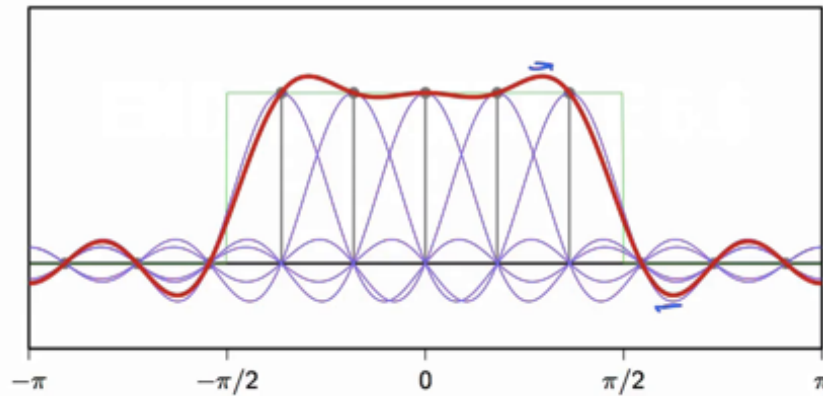
notes

summary

2m 25s



Frequency sampling: frequency response



So these are good methods to know in order to quickly try something out when we're faced with a filtering problem. But definitely they are not optimal and they leave a lot to be desired in terms of the fine control that we can have over the maximum error.

notes

summary

2m 37s

