Discrete Fourier Transform and filters

Eugeniy E. Mikhailov

The College of William & Mary

Lecture 24

DFT vs Matlab FFT

DFT

$$y_k = \frac{1}{N}\sum_{n=0}^{N-1} c_n \exp(i\frac{2\pi(k-1)n}{N}) \quad \text{inverse Fourier transform}$$

$$c_n = \sum_{k=1}^{N} y_k \exp(-i\frac{2\pi(k-1)n}{N}) \quad \text{Fourier transform}$$

$$n = 0, 1, 2, \cdots, N-1$$

Matlab FFT

$$y_k = \frac{1}{N}\sum_{n=1}^{N} c_n \exp(i\frac{2\pi(k-1)(n-1)}{N}) \quad \text{inverse Fourier transform}$$

$$c_n = \sum_{k=1}^{N} y_k \exp(-i\frac{2\pi(k-1)(n-1)}{N}) \quad \text{Fourier transform}$$

$$n = 1, 2, \cdots, N$$

So do DFT $\rightarrow$ Matlab FFT is equivalent of $n \rightarrow n+1$ and vice versa

Warning about notation

Since c_0 has a special meaning of a DC component of the signal.
I will always use the DFT notation unless mentioned otherwise.
People often denote the forward Fourier transform as $\mathcal{F}$ so

$$Y = \mathcal{F}y$$

So Y is the spectrum of the signal y
Inverse Fourier transform is denoted as $\mathcal{F}^{-1}$

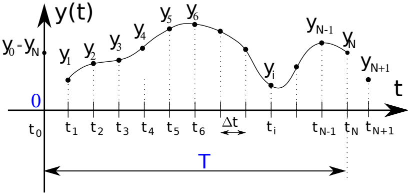
$$y = \mathcal{F}^{-1}Y$$

Instead of using c_n coefficient we refer in this notation to Y_n

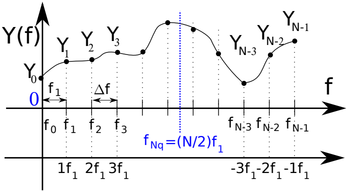
Sampling rate and important physics relationship

Since for DFT we need to have equidistant points and signal repeats itself. We consider signals which start at time 0 and take N points. To deduce the time of the data point we just multiply it's index by the time spacing Δt.

Time series



Spectrum



Sampling rate is defined as $f_s = 1/\Delta t = f_1 N$ and period $T = N\Delta t$.
y_i is taken at time $t_i = i\Delta t = i/f_s$, $y_{i+N} = y_i$.
$Y_{i+N} = Y_i$.
In matlab fft Y_n has the frequency $f_n = f_1(n-1) = f_s(n-1)/N$.

Nyquist frequency

Provided that we have N data point taken with sampling rate f_s what is the maximum frequency which we can expect to see in our spectrum?
Naively, we can say $(N-1) * f_1 \approx f_s$ since in spectrum all points are separated by fundamental frequency $f_1 = 1/T = f_s/N$
However recall that $Y_{N-n} = Y_{-n}$ i.e the higher half of the vector Y contains negative frequency. So at max we can hope to obtain spectrum with the highest frequency smaller than

Nyquist frequency

$$F_{Nq} = f_1 \frac{N}{2} = \frac{f_s}{2}$$

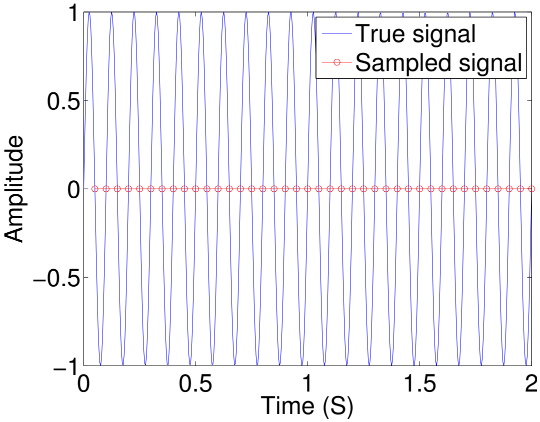
Nyquist criteria

$$f_s > 2f_{signal}$$

You must sample your signal faster than twice the highest frequency component of it. I.e. Nyquist frequency of you sample should be $>$ than the highest signal frequency.

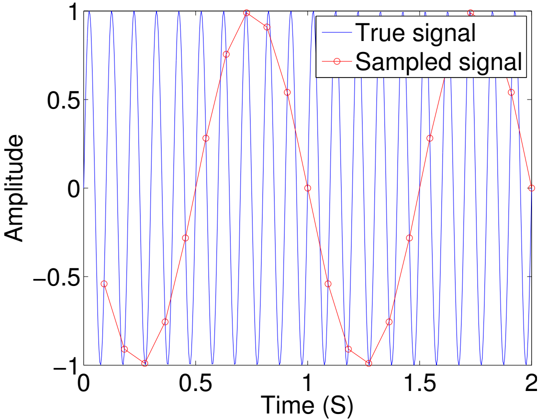
Aliasing: wrong/slow sampling frequency

Sampling with
$f_s = 2f_{signal}$
i.e.
$f_{Nq} = f_{signal}$
Sampled signal
appeared to be DC



Aliasing: too slow sampling frequency - reflection

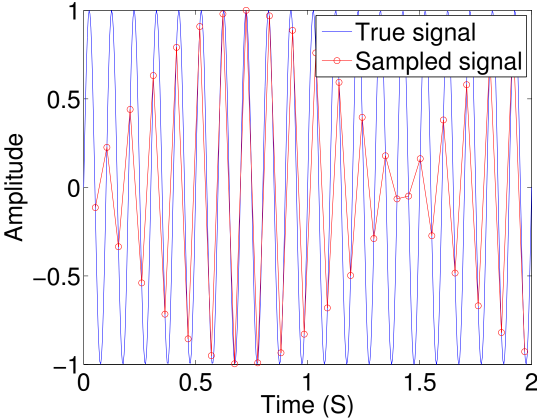
Under sampling
$f_s = 1.1 f_{signal}$
Sampled signal seems to be lower frequency.



This is case of reflection/folding when frequency higher than the Nyquist frequency appears to be negative and slower one.
$f_{signal} \rightarrow (f_{signal} - 2f_{Nq})$

Aliasing: too slow sampling frequency - ghosts

Under sampling
$f_s = 1.93 f_{signal}$
Sampled signal seems to be very different



This is also a case of reflection/folding when frequency higher than the Nyquist frequency appears to be negative and slower one.

DFT filters

Once you get a signal you can filter unwanted components out of it. The recipe is the following

- sample the signal
- calculate FT (fft)
- have a look at the spectrum and decide which components are unwanted
- apply filter which attenuate unwanted frequency component (remember that if you attenuate the component of the frequency f by g_f you need to attenuate the component at $-f$ by g_f^*.
- calculate inverse FT (ifft) of the filtered spectrum
- repeat if needed

Applications

- Noise reduction
- Compression