# FFT

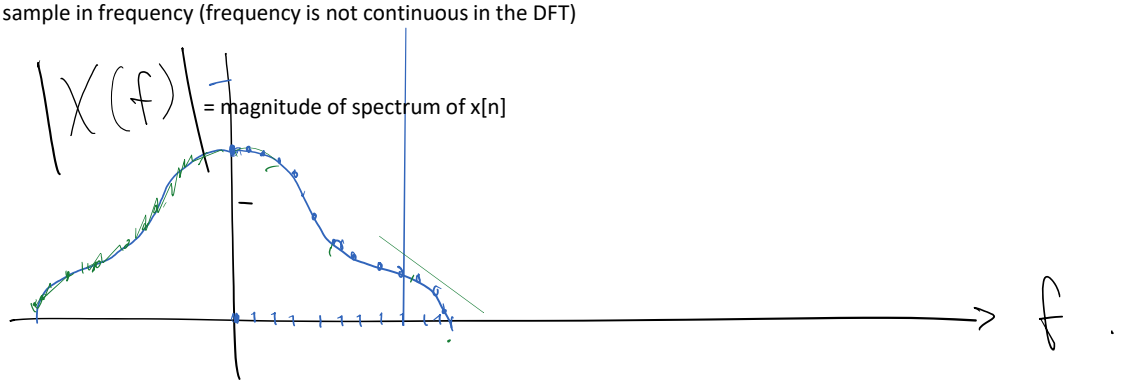
Monday, February 28, 2022 3:26 PM

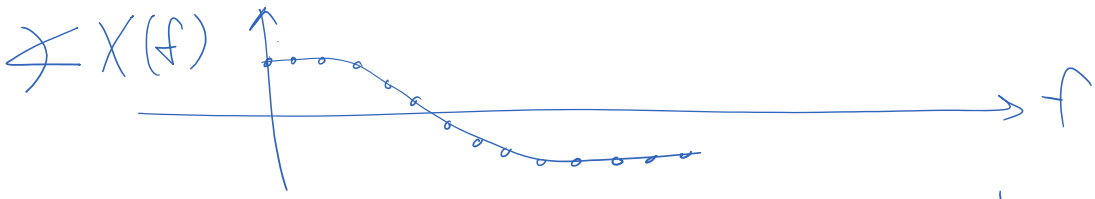
FFT = fast Fourier transform

Computationally efficient way to compute the DTFT

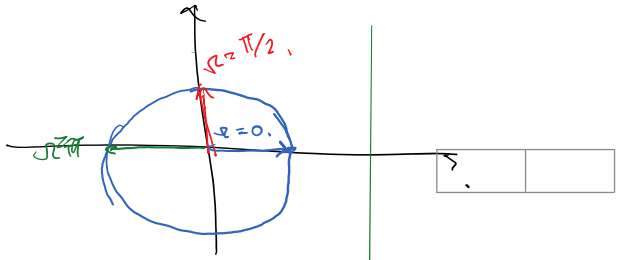
FFT is actually the Discrete Fourier Transform DFT which is an approximation of the DTFT

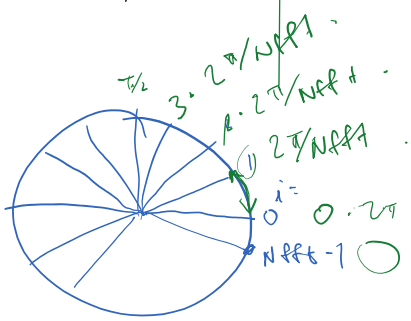
- We don't have an infinite number of points for x (or could not calculate the summation to n = infinity)
- Need to sample in frequency (frequency is not continuous in the DFT)




specify $N_{fft}$ = total # of frequencies @ which to sample spectrum.




[0 : Nfft-1] * 2π/Nfft

Matlab command fft

Xf = fft(x, Nfft) # of frequencies at which to compute DFT.

Xf = fft(x, Nfft) # of frequencies at which to compute

DFT of x[n]

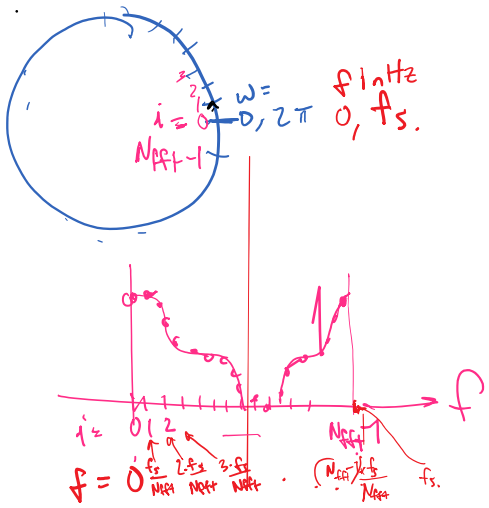
input signal that we want spectrum of

To view spectrum

f = [0 : (Nfft-1)] * (fs/Nfft).

plot(f, abs(Xf))

w = 0

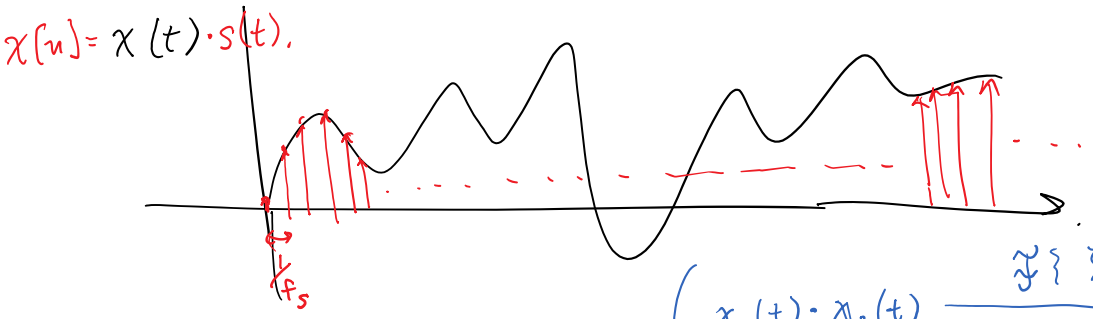


# DTFT - mathematical formulation

Wednesday, March 2, 2022 3:14 PM

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n] \cdot e^{j\Omega n}$$

$x[n] = x(t) \cdot s(t)$.



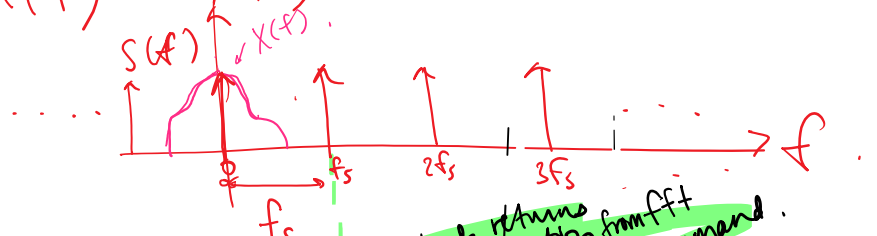

$$\left( x_1(t) \cdot x_2(t) \xrightarrow{\mathcal{F}\{\}} X_1(f) * X_2(f). \right.$$

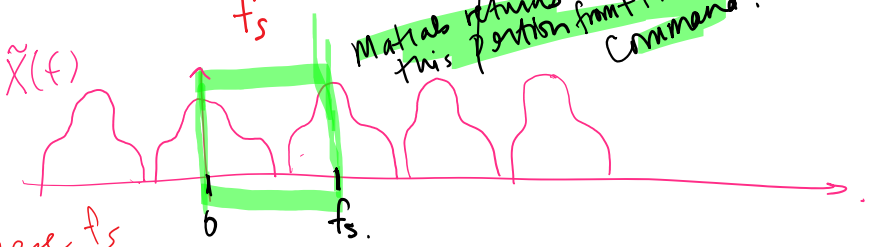
$$x_1(t) \xrightarrow{\mathcal{F}\{\}} X_1(f)$$

$$x_2(t) \xrightarrow{\mathcal{F}\{\}} X_2(f).$$

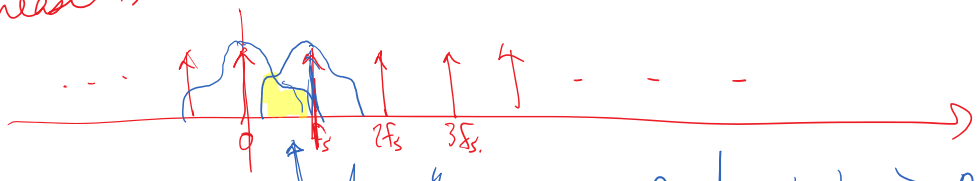
$$X(e^{j\Omega}) = X(j\omega) * S(j\omega)$$

$$\tilde{X}(f) = X(f) * S(f)$$




If I Decrease $f_s$




"aliasing" → $f_s$ must be > 2·(bandwidth)