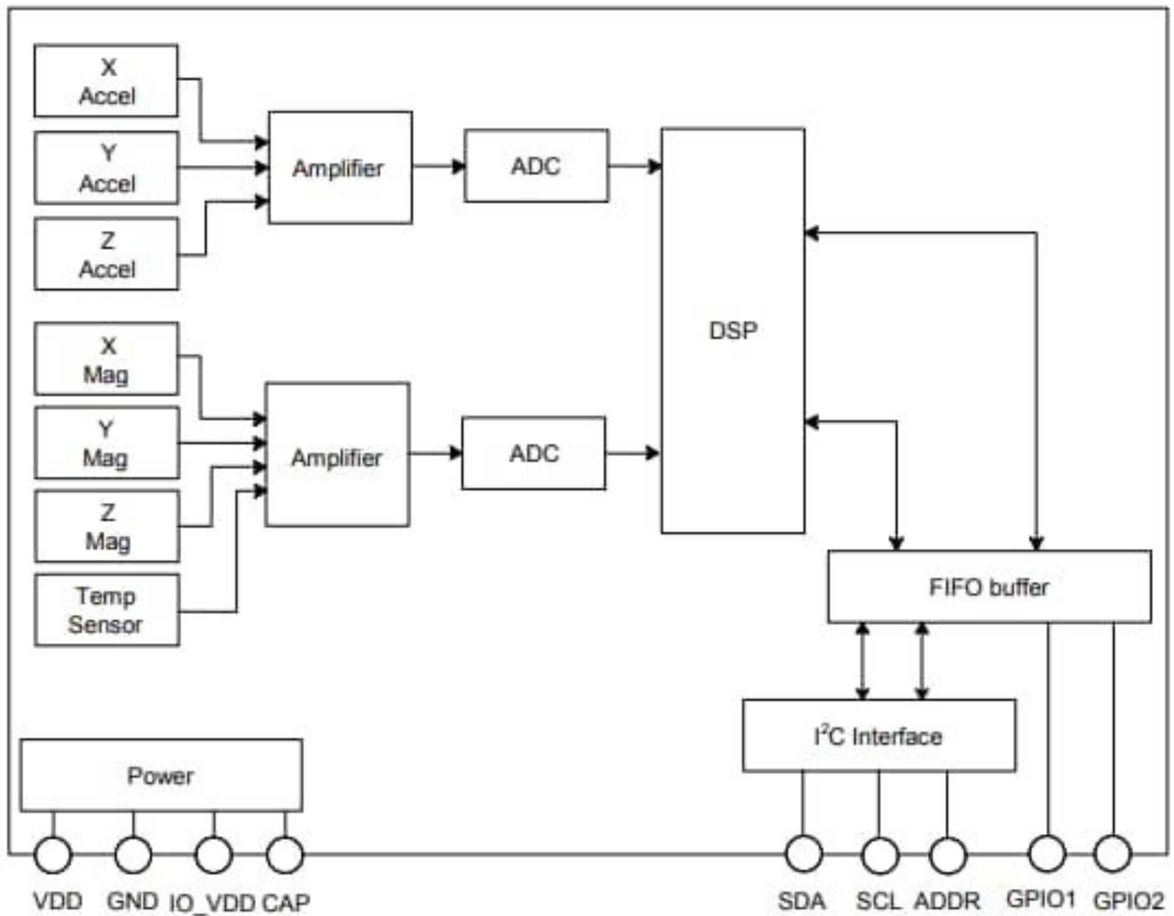
Digital Signal Processing Fundamentals

DSP Basics

- ADC & DAC
- Signal Basics (Sine Wave)
- ADC: Sampling - What's proper sampling rate?
  - Nyquist sampling rate
  - Aliasing
- Discrete Fourier Transform (DFT)
  - How to do?
  - DFT and its relationship with other transforms
  - DFT properties: symmetry, resolution
  - DFT examples
  - Using DFT in Python
- Energy vs. Power

Sensor Internals

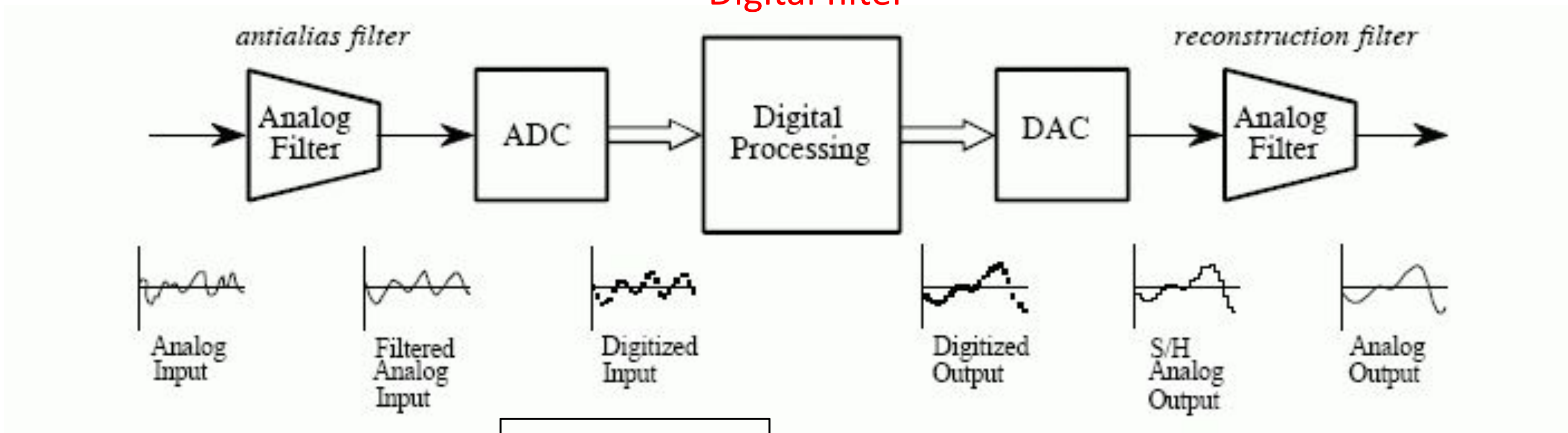


http://web.mit.edu/6.111/www/f2017/handouts/L13.pdf

ADC & DAC

- Block diagram of a DSP system

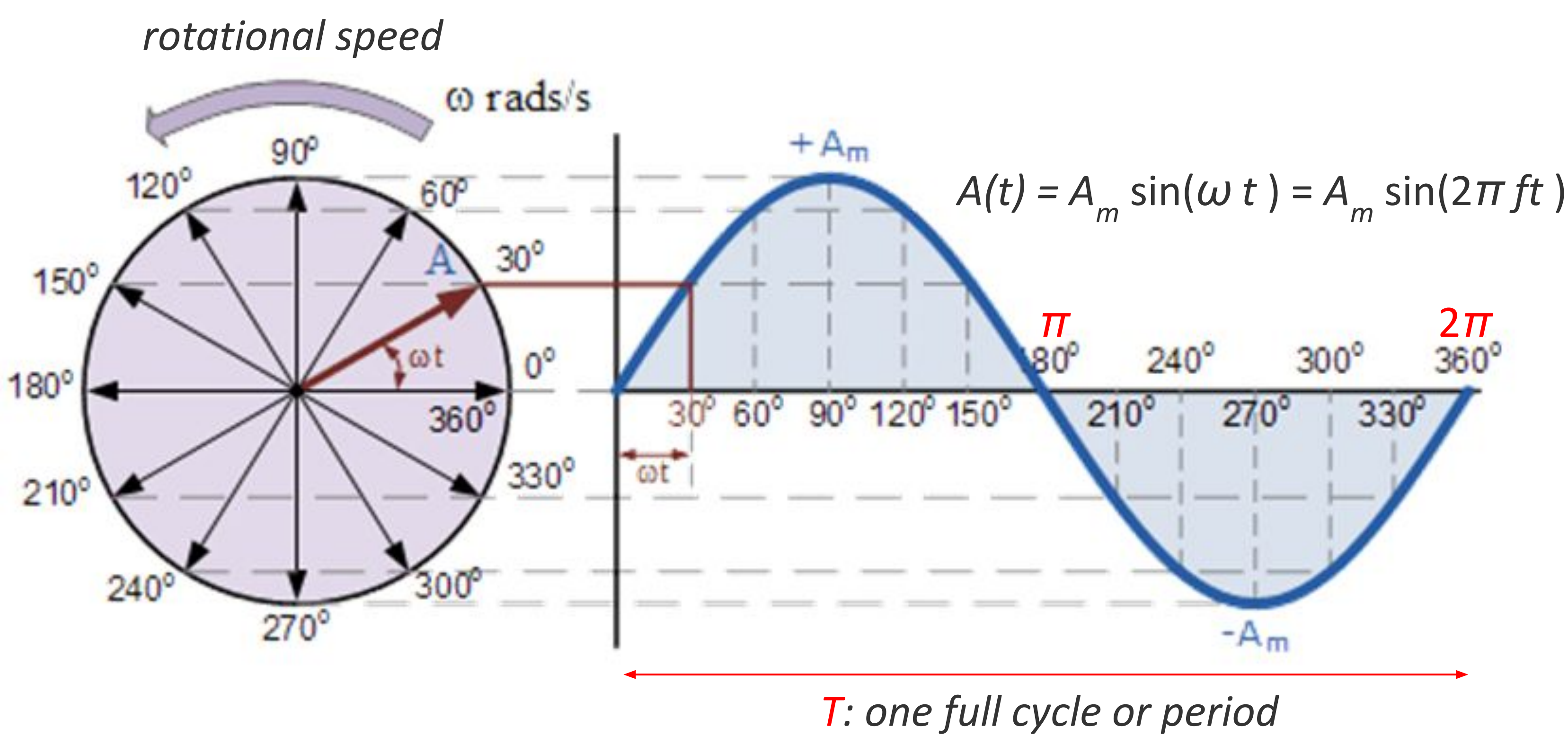
Digital filter



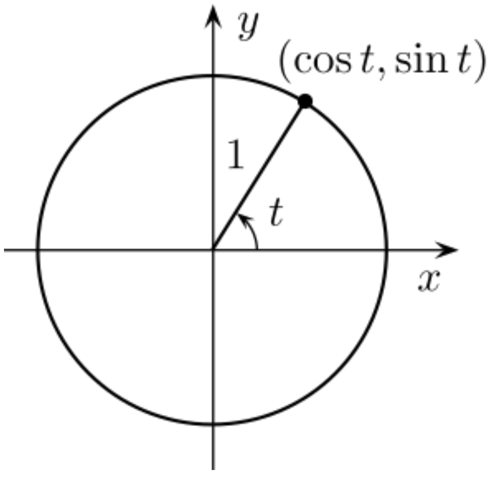
Sampling
Quantization

Sample an input signal for every T seconds, which is represented using k bits

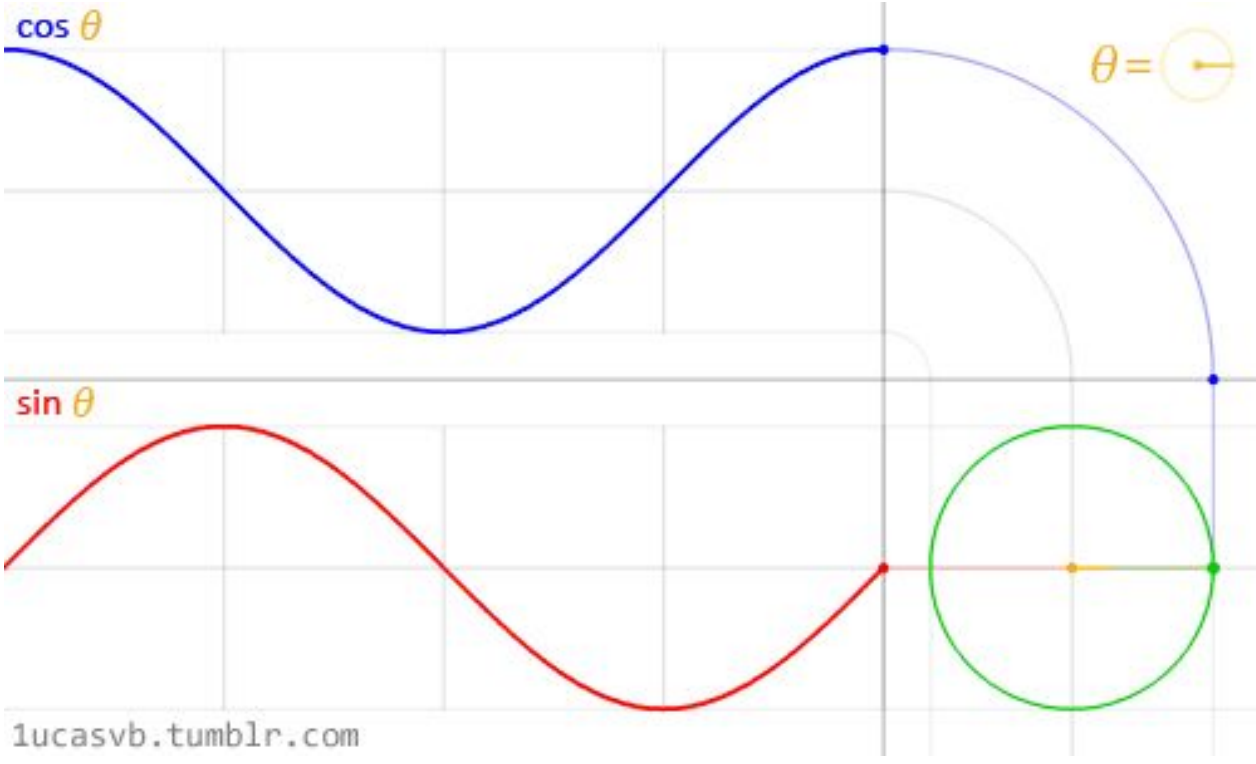
Signal Basics



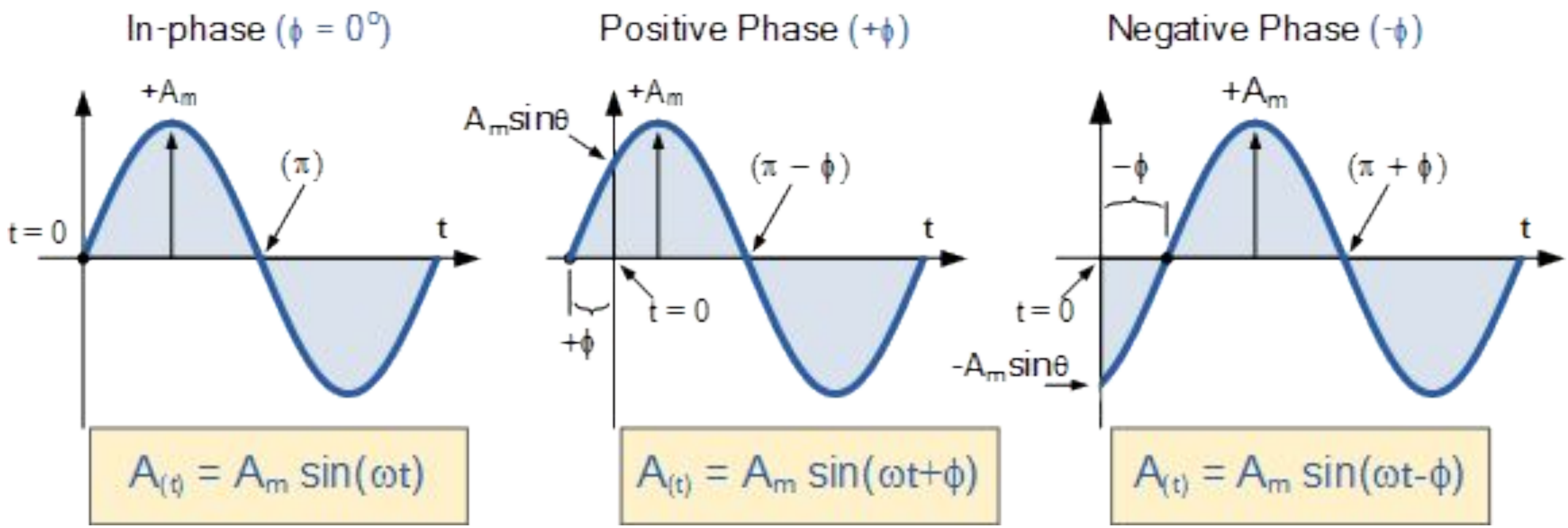
https://www.electronics-tutorials.ws/accircuits/phase-difference.html



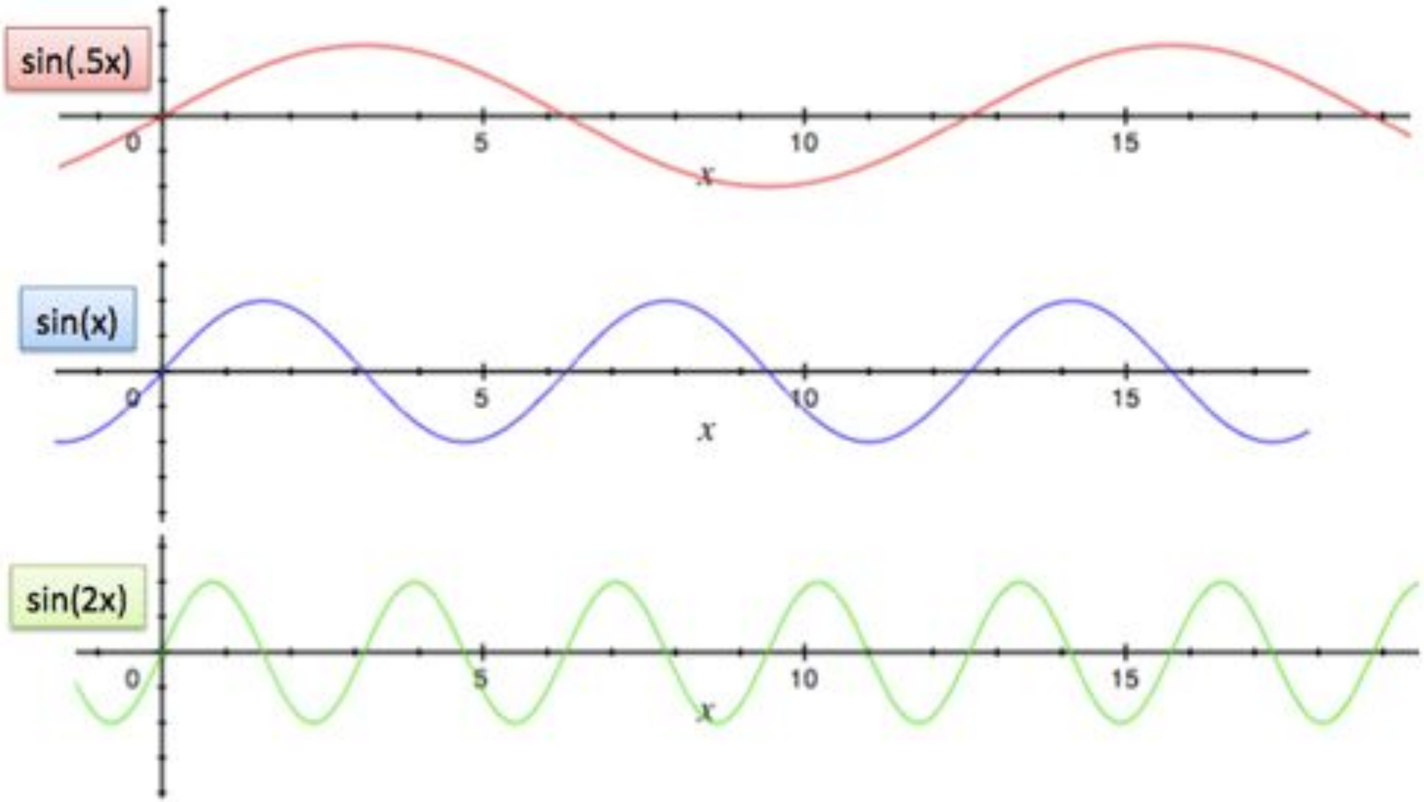
https://en.wikipedia.org/wiki/Negative_frequency



http://1ucasvb.tumblr.com



https://www.electronics-tutorials.ws/accircuits/phase-difference.html



https://betterexplained.com/articles/intuitive-understanding-of-sine-waves/

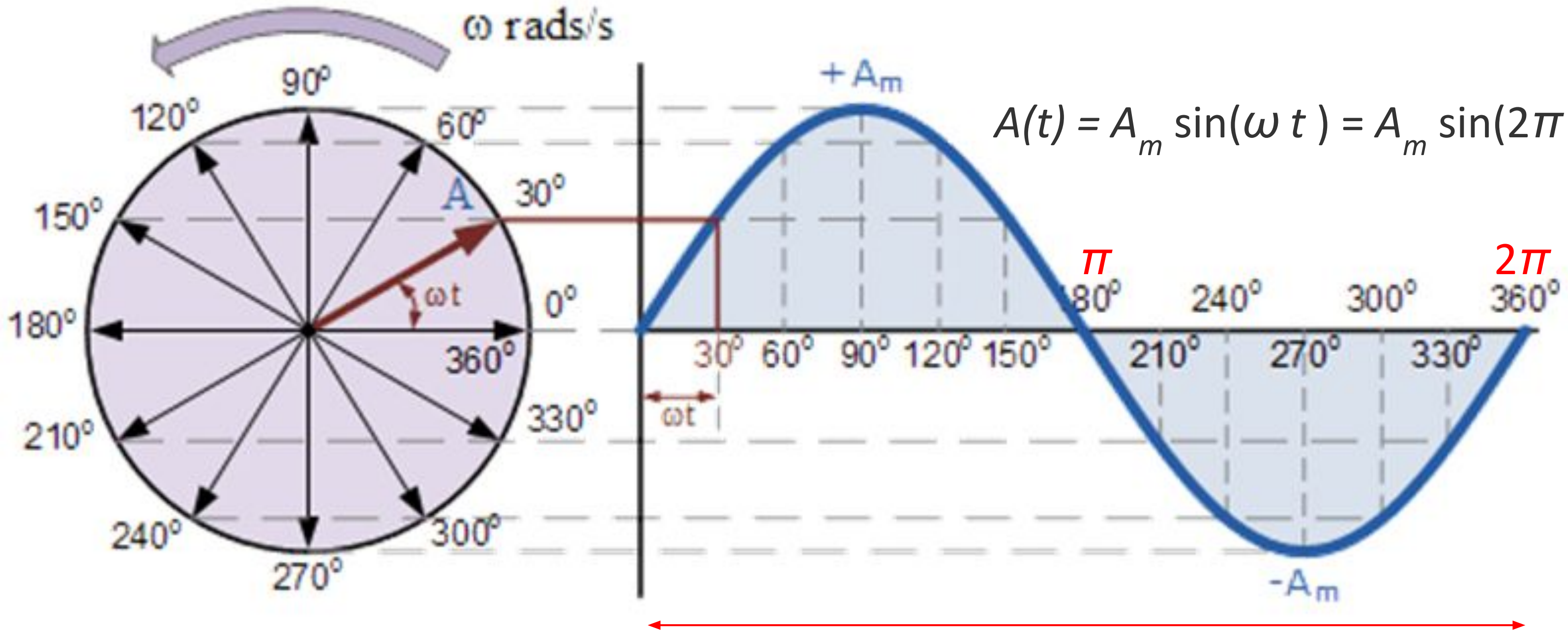
Signal Basics

$\boldsymbol{\omega = 2\pi f \Leftrightarrow T = 1 / f}$

$\omega = 2\pi / T \Leftrightarrow T = 2\pi / \omega \Leftrightarrow T^*\omega = 2\pi$

$f = 1 / T = \omega / 2\pi$

rotational speed

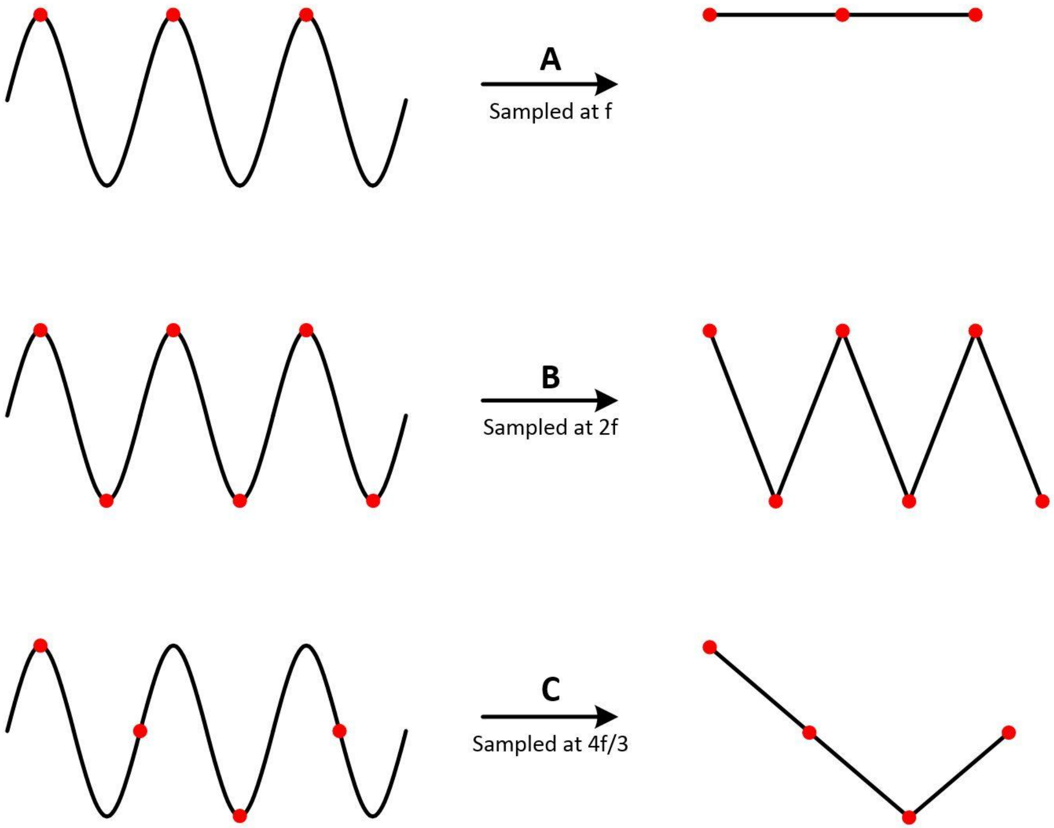


$A(t) = A_m \sin(\omega t) = A_m \sin(2\pi f t)$

T: one full cycle or period

https://www.electronics-tutorials.ws/accircuits/phase-difference.html

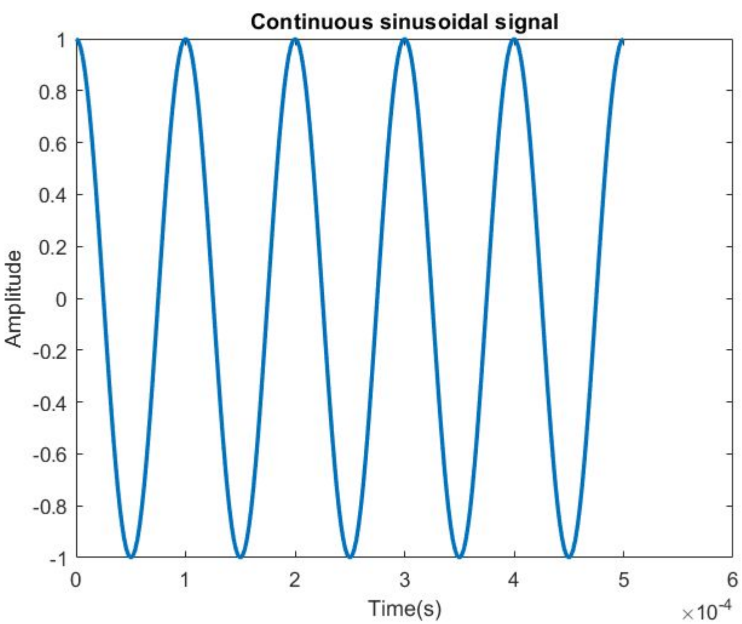
ADC: Sampling – What's proper sampling rate?



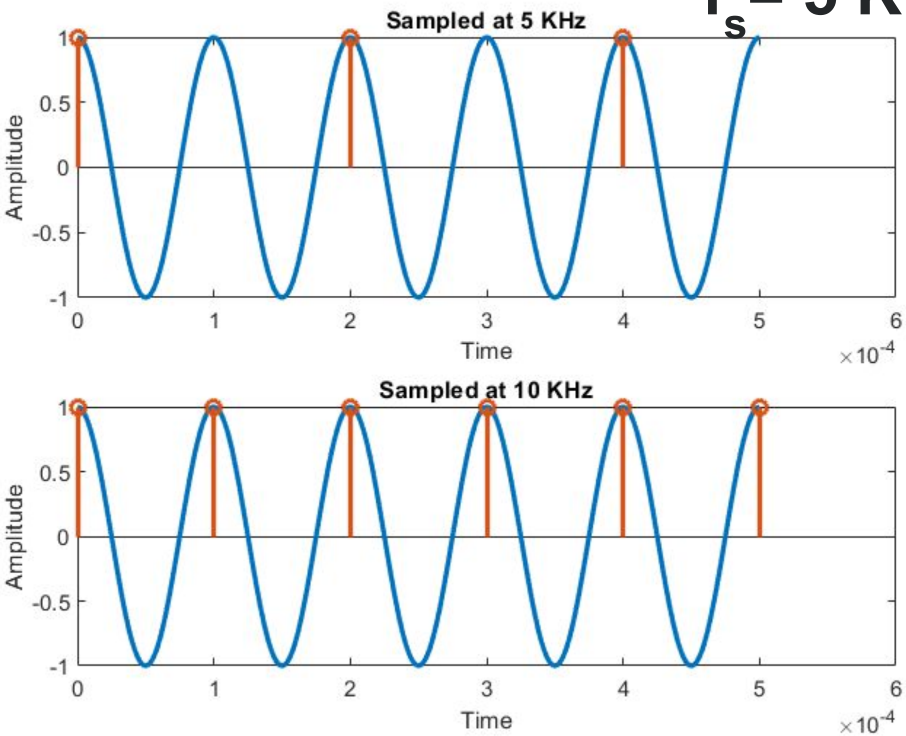
https://psychology.stackexchange.com/questions/18237/what-happens-at-sampling-rates-lower-or-higher-that-the-nyquist-rate

ADC: Sampling – What's proper sampling rate?

B=10 KHz



f_s= 5 KHz

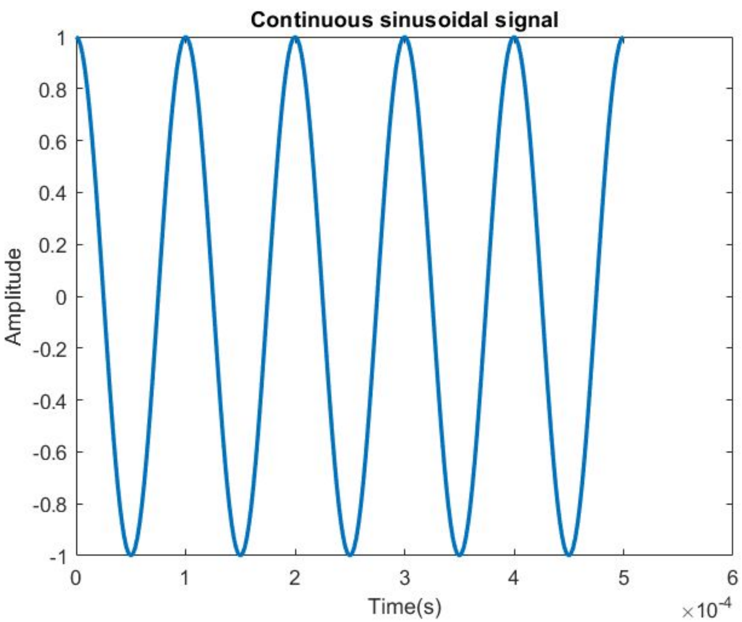


f_s= 10 KHz

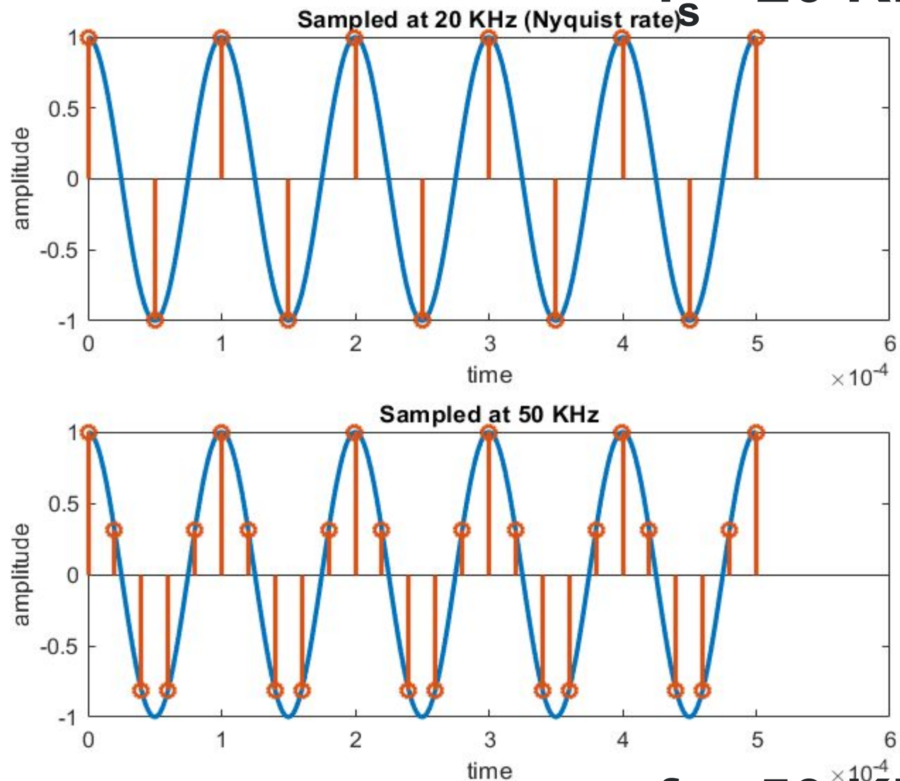
https://psychology.stackexchange.com/questions/18237/what-happens-at-sampling-rates-lower-or-higher-that-the-nyquist-rate

ADC: Sampling – What's proper sampling rate?

B=10 KHz



f_s = 20 KHz

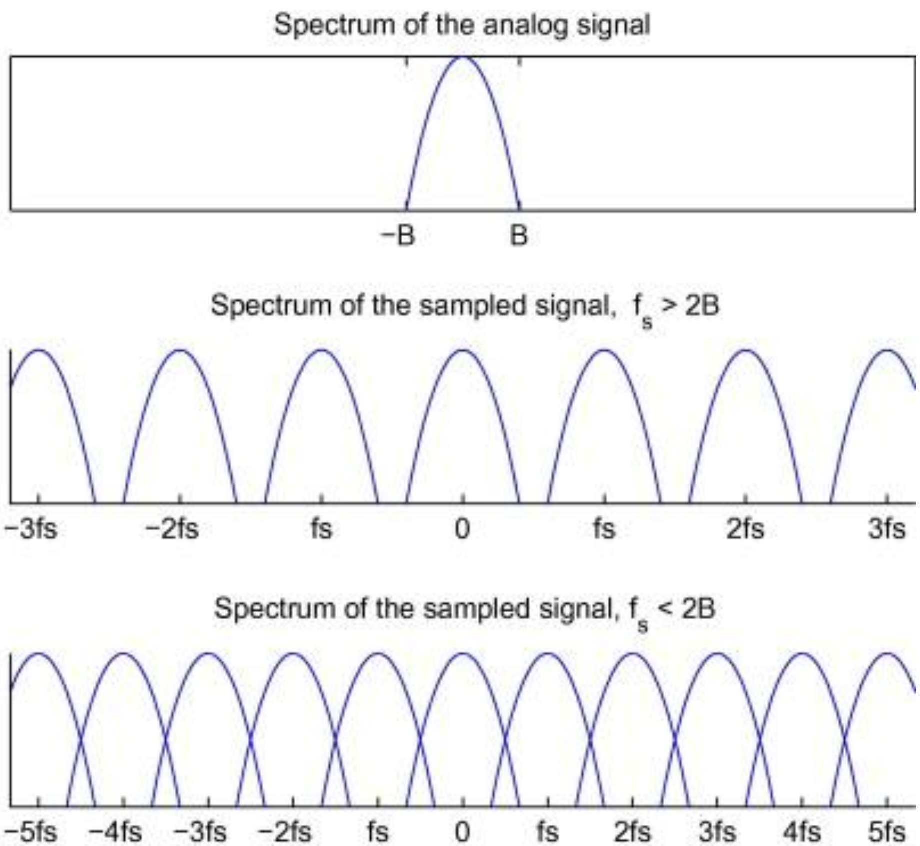


f_s = 50 KHz

Roughly 2 times of original signal!

https://psychology.stackexchange.com/questions/18237/what-happens-at-sampling-rates-lower-or-higher-that-the-nyquist-rate

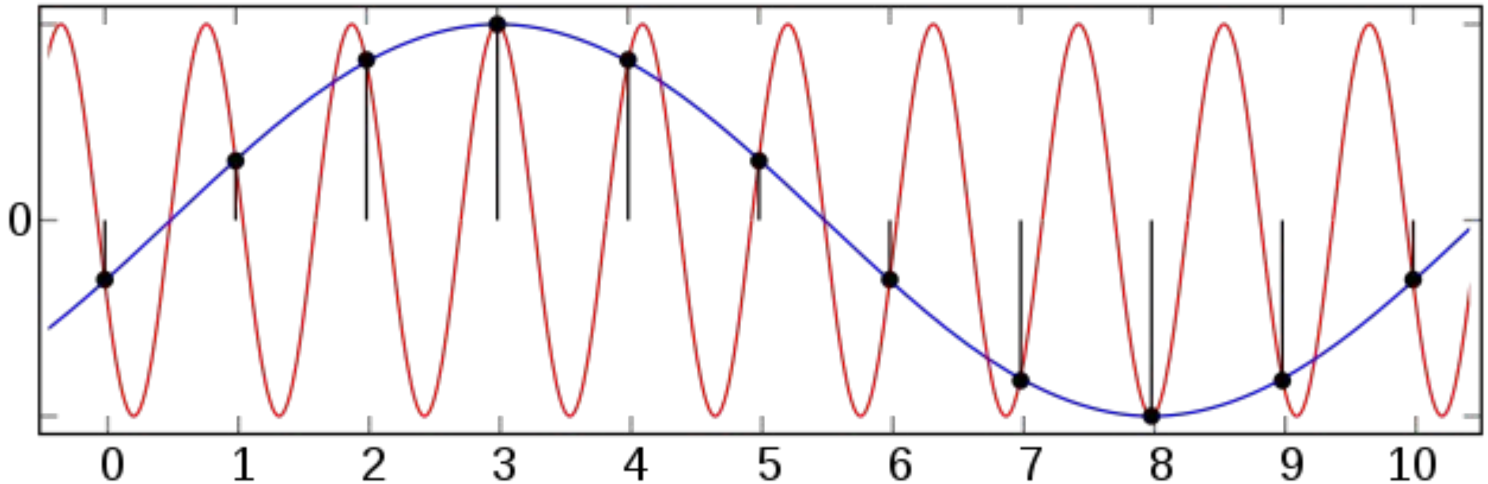
ADC: Sampling – What's proper sampling rate?



Why are there replicas?
It's the artifact of sampling!
(due to the multiplication of impulse trains)

https://www.osapublishing.org/oe/fulltext.cfm?uri=oe-24-5-4842&id=336753
For more information about "replicas" please watch: https://www.youtube.com/watch?v=mxdf_fSE2Gg

What happens if sampling rate is too low? Aliasing



Oops, we're capturing the blue signal, instead of the red signal

*This phenomenon of sinusoids changing frequency during sampling is called **aliasing**. Just as a criminal might take on an assumed name or identity (an alias), the sinusoid assumes another frequency that is not its own. Aliased signals are within the rage of sampling rate!!*

https://en.wikibooks.org/wiki/Analog_and_Digital_Conversion/Nyquist_Sampling_Rate

Wait! We’re dealing w/ digital signals!

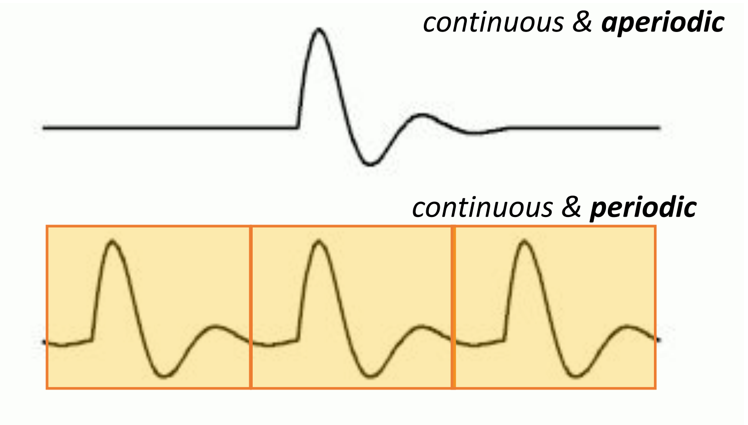
Why do we need to care about aliasing?

- When you have high resolution digital signals, you may want to perform “under-sampling” (say to improve computation efficiency)
- If you’re doing under-sampling, you need to remove “higher frequency” components than “under-sampling” frequency

Transforms

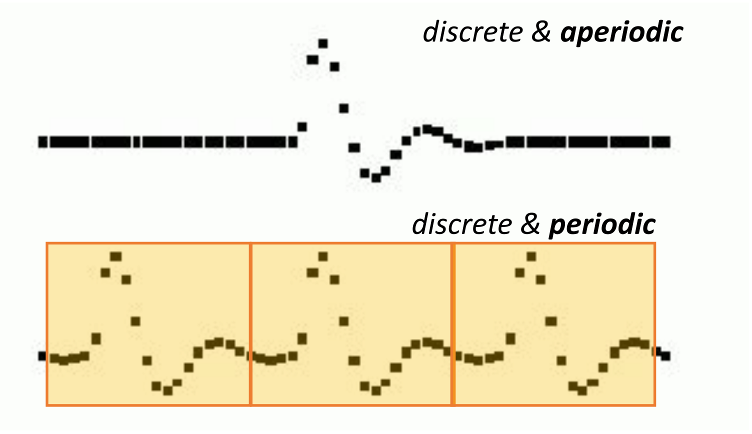
Fourier Transform
*signals that are **continuous** and **aperiodic***

Fourier Series
signals that are continuous and periodic



Discrete Time Fourier Transform
*signals that are **discrete** and **aperiodic***

Discrete Time Fourier Series
signals that are discrete and periodic



https://www.dspguide.com/ch8/1.htm

Correlation

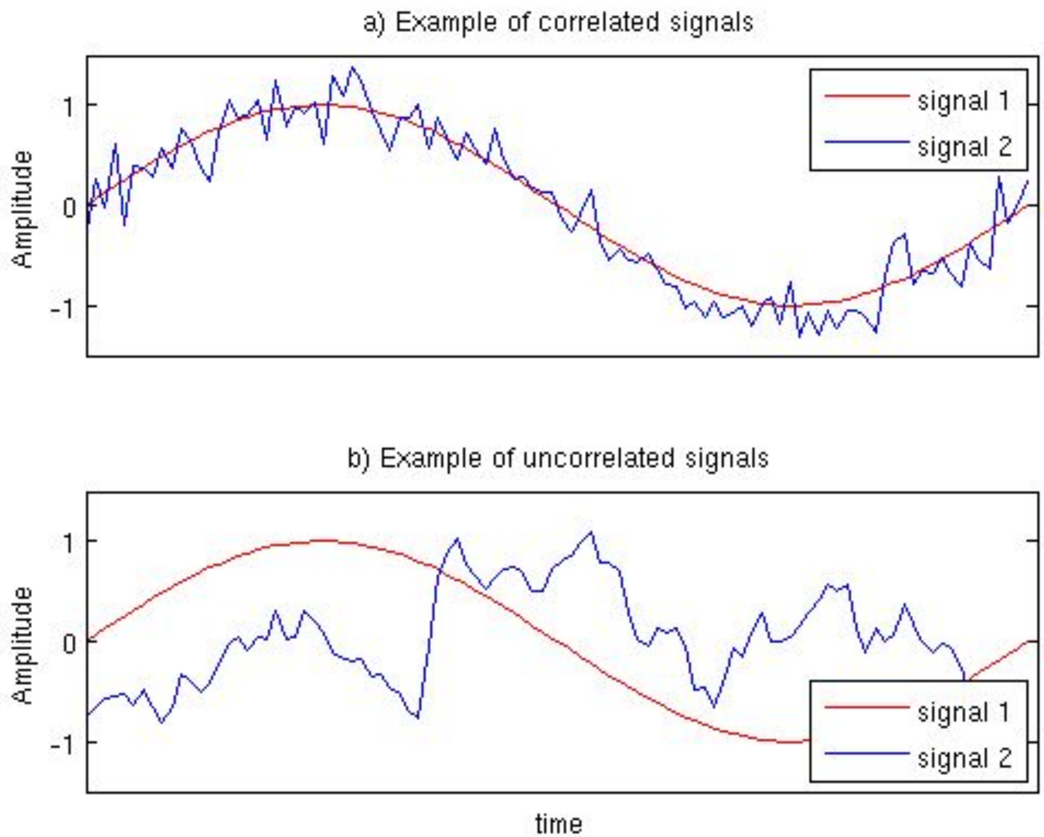
- Correlation between two signals: x(i) and y(i)

$$\sum_{i=0}^{N} x(i)y(i)$$

Correlation

$$\sum_{i=0}^{N} x(i)y(i)$$

- Correlation between two signals: x(i) and y(i)



http://practicalcryptography.com/miscellaneous/machine-learning/intuitive-guide-discrete-fourier-transform/

DFT: Discrete Fourier Transform

$$\boxed{x(n)} \xrightarrow{\textit{DFT}} \boxed{X(k)}$$

N samples — *N frequency elements*

- $$X(k) = \sum_{n=0}^{N-1} x(n) e^{-i2\pi kn/N}$$
  - *Use Euler's formula:* $e^{-i\theta} = \cos\theta - i\sin\theta$
  - *Here k ranges from 0 to N-1*
- $$X(k) = \sum_{n=0}^{N-1} x(n)\left(\cos\left(2\pi kn/N\right) - i\sin\left(2\pi kn/N\right)\right)$$
- $$X(k) = \boxed{\sum_{n=0}^{N-1} x(n)\cos\left(\frac{2\pi kn}{N}\right)} - i\boxed{\sum_{n=0}^{N-1} x(n)\sin\left(\frac{2\pi kn}{N}\right)}$$

Correlation:
∑x(n)y(n)
y(n) = cos(Ωkn)

Basis function!
cos(Ωkn)

http://practicalcryptography.com/miscellaneous/machine-learning/intuitive-guide-discrete-fourier-transform/

Fourier Series

- Three different representations

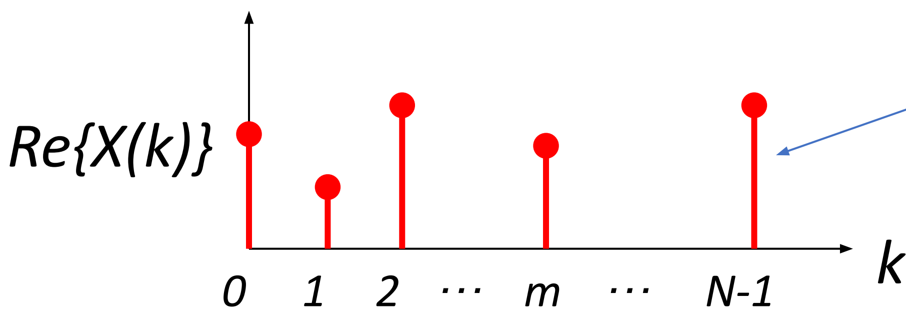
Trigonometry	$x(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos k\omega_0 t + \sum_{k=1}^{\infty} b_k \sin k\omega_0 t$	$a_0 = \frac{1}{T}\int_T x(t)\,dt$ $a_k = \frac{2}{T}\int_T x(t)\cos k\omega_0 t\,dt$ $b_k = \frac{2}{T}\int_T x(t)\sin k\omega_0 t\,dt$	$a_0 = c_0 = X_0$ $a_k = c_k \cos\phi_k = X_k + X_{-k}$ $b_k = -c_k \sin\phi_k = j(X_k - X_{-k})$
Simplified Trigonometry	$x(t) = \sum_{k=0}^{\infty} c_k \cos(k\omega_0 t + \phi_k)$	$c_0 = a_0,\ \phi_0 = 0$ $c_k = \sqrt{a_k^2 + b_k^2}$ $\phi_k = -\tan^{-1}\left(\frac{b_k}{a_k}\right)$	$c_0 = X_0$ $c_k = 2\lvert X_k\rvert$ $\phi_k = \angle X_k$
Exponential Form	$x(t) = \sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t}$	$X_k = \frac{1}{T}\int_T x(t) e^{-jk\omega_0 t}\,dt$	$X_0 = a_0 = c_0$ $X_k = \frac{c_k}{2} e^{j\phi_k} = \frac{1}{2}(a_k - jb_k)$ $X_{-k} = X_k^*$

DFT: Discrete Fourier Transform

$$x(n) \xrightarrow{\textit{DFT}} X(k)$$

N samples — *N frequency elements*

$$X(k) = \underbrace{\boxed{\sum_{n=0}^{N-1} x(n)\cos\left(\frac{2\pi kn}{N}\right)}}_{\text{real}} - i\,\underbrace{\boxed{\sum_{n=0}^{N-1} x(n)\sin\left(\frac{2\pi kn}{N}\right)}}_{\text{imaginary}}$$

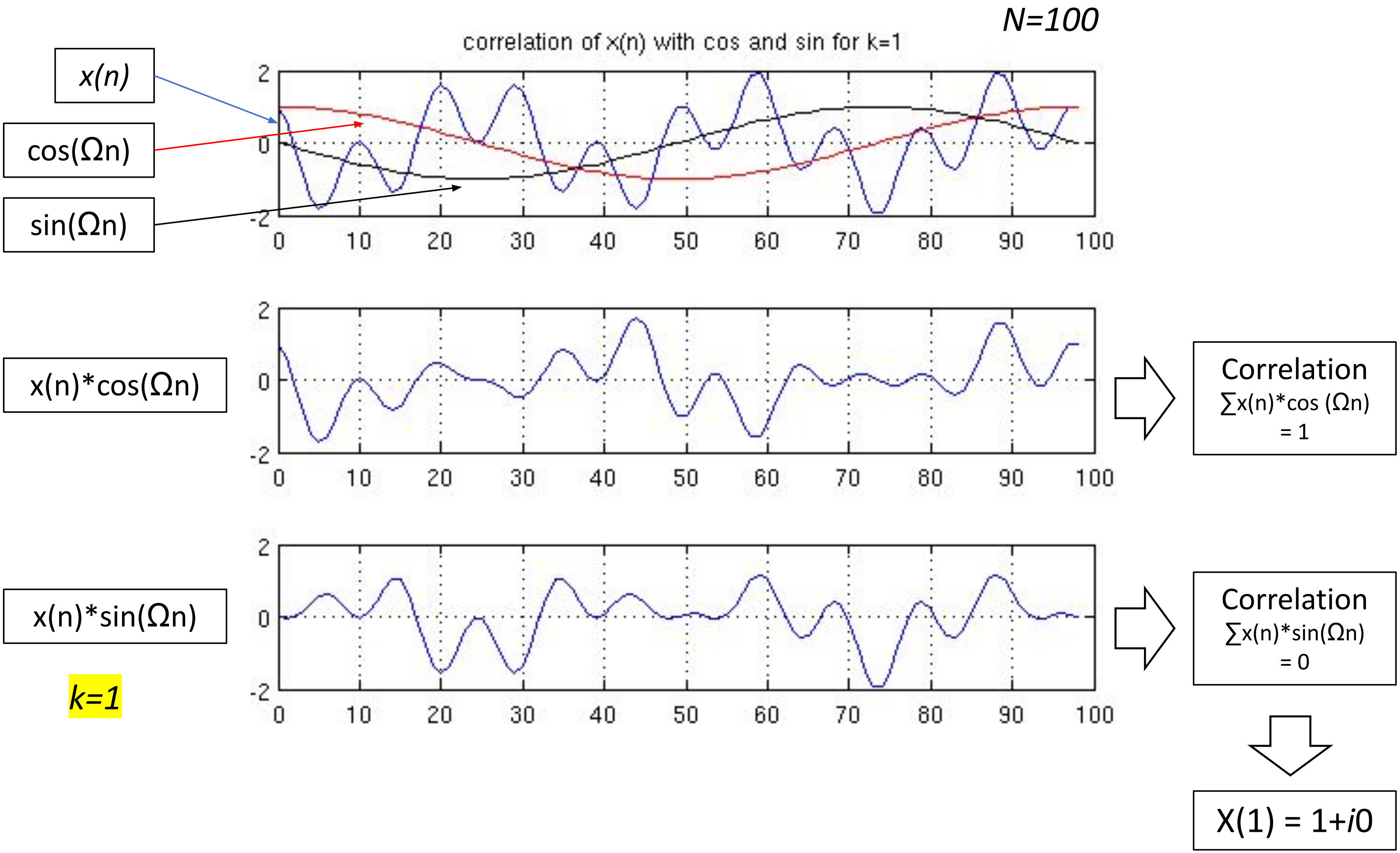


Correlation value =
$X(k) = \sum x(n)*\cos(\Omega kn)$

Each slot = f_s / N
where f_s is the sampling frequency

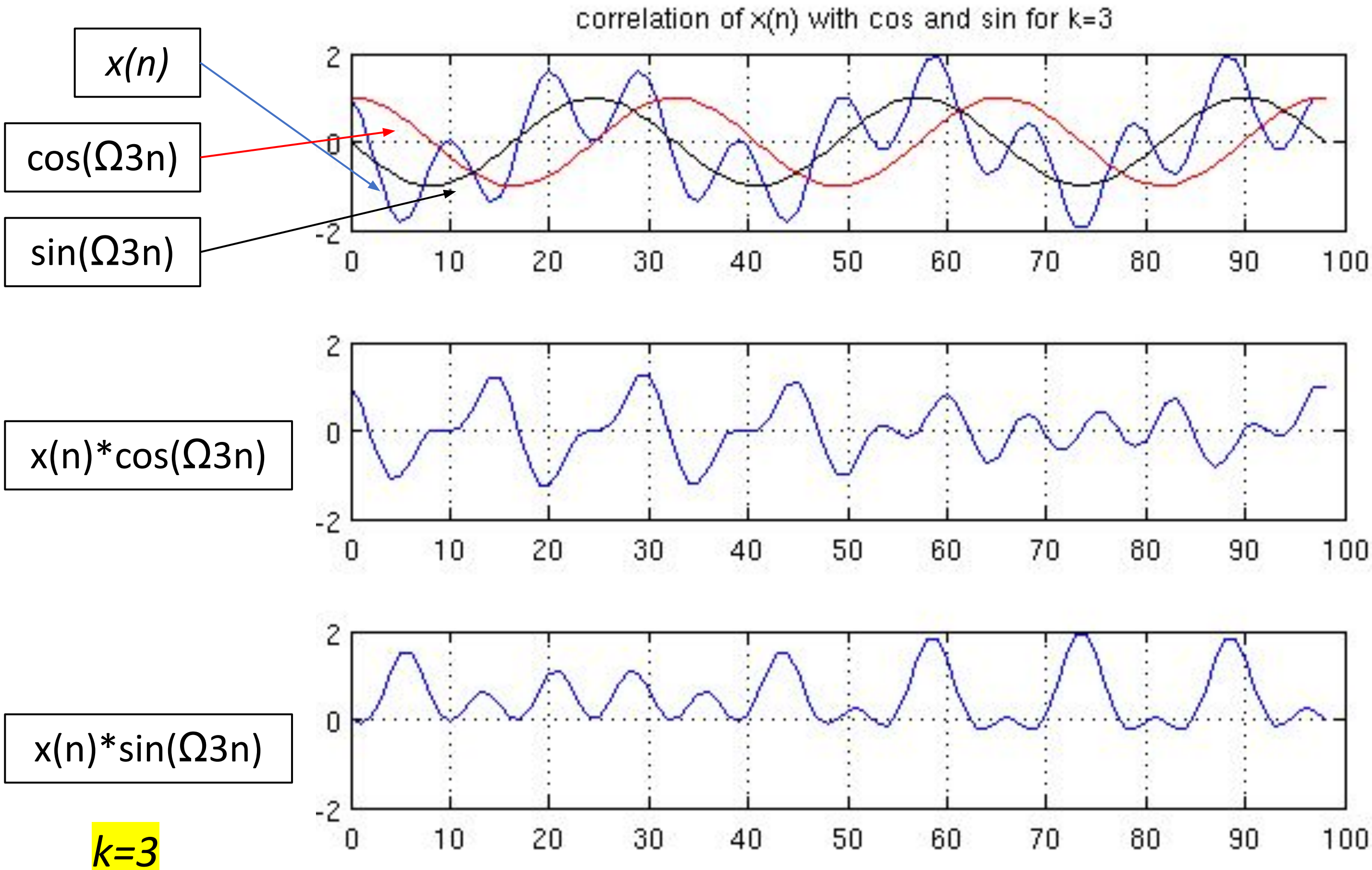
http://practicalcryptography.com/miscellaneous/machine-learning/intuitive-guide-discrete-fourier-transform/

DFT: Discrete Fourier Transform



http://practicalcryptography.com/miscellaneous/machine-learning/intuitive-guide-discrete-fourier-transform/

DFT: Discrete Fourier Transform



Correlation
∑x(n)*cos (Ω3n)
= 0

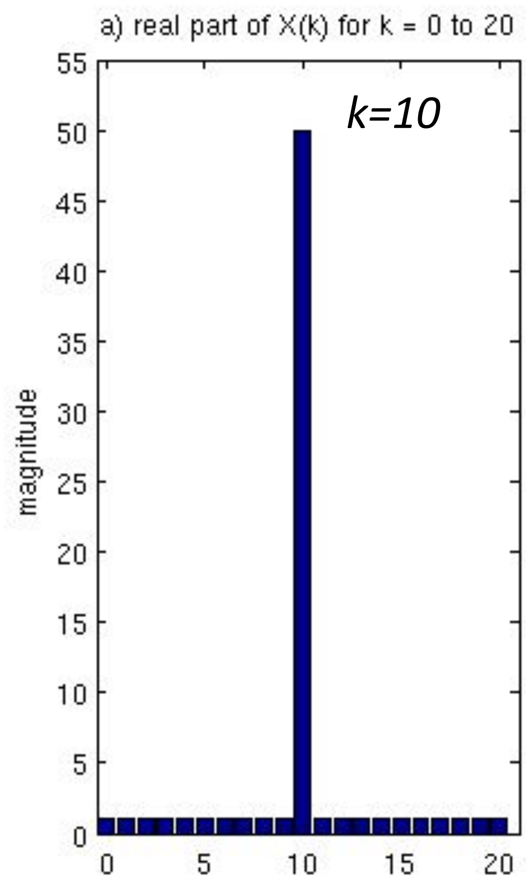
Correlation
∑x(n)*sin(Ω3n)
= 49

X(3) = 0+*i*49

http://practicalcryptography.com/miscellaneous/machine-learning/intuitive-guide-discrete-fourier-transform/

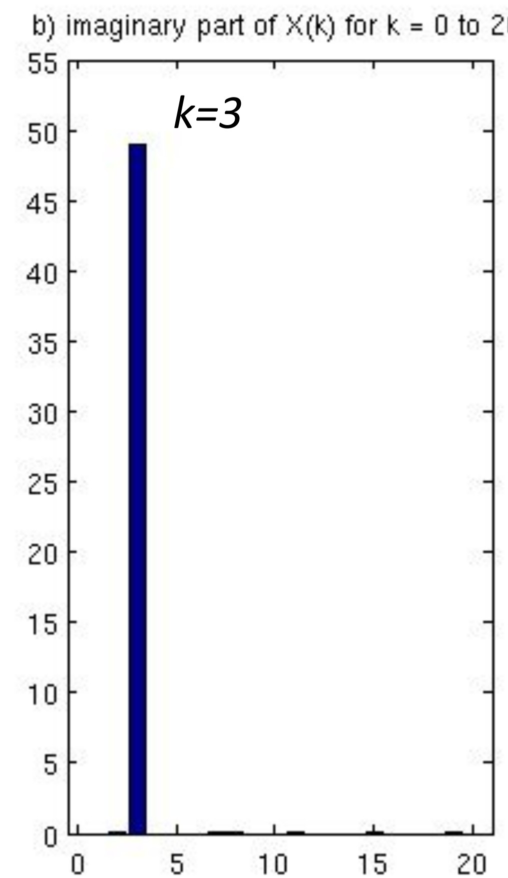
DFT: Discrete Fourier Transform

Re{X(k)}



k

Im{X(k)}



k

$$f = \frac{k \times \mathrm{fs}}{N}$$

Conversion to frequency

http://practicalcryptography.com/miscellaneous/machine-learning/intuitive-guide-discrete-fourier-transform/

DFT: Discrete Fourier Transform

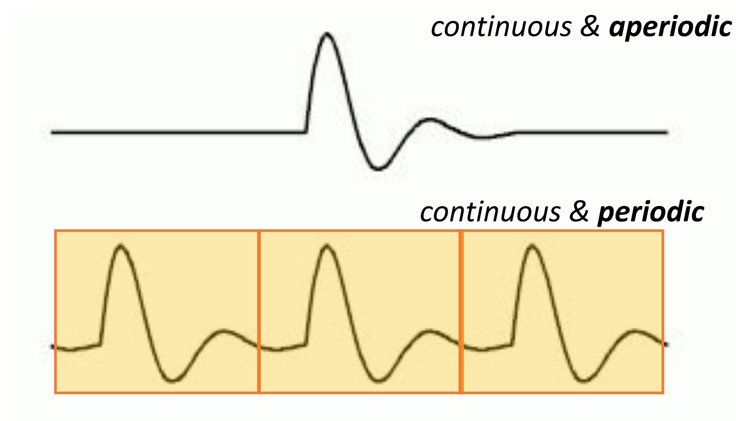
DFT $$X(k) = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

Inverse DFT $$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}$$

DFT and its relationship with other transforms

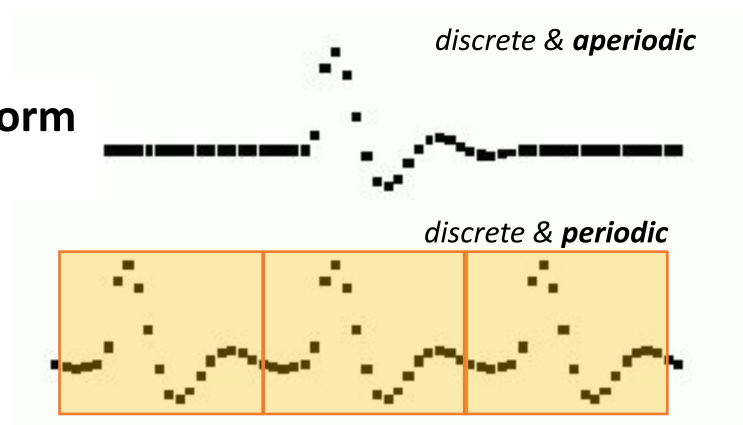
FT: Fourier Transform
signals that are continuous and aperiodic

FS: Fourier Series
signals that are continuous and periodic



DTFT: Discrete Time Fourier Transform
signals that are discrete and aperiodic

DTFS: Discrete Time Fourier Series
signals that are discrete and periodic

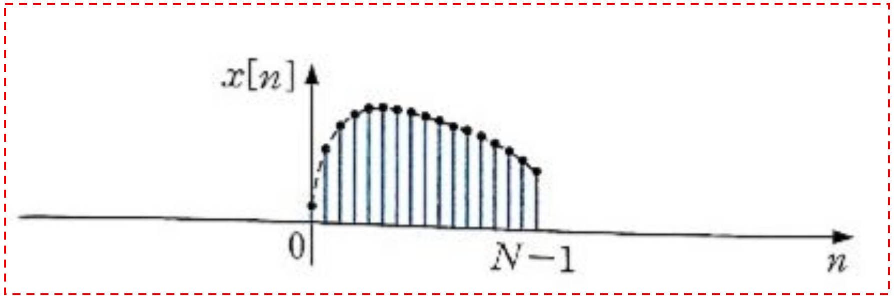


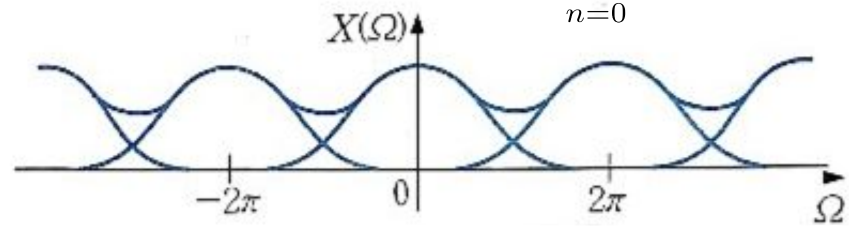
https://www.dspguide.com/ch8/1.htm

DFT and its relationship with other transforms

DTFT: Discrete Time Fourier Transform

$$DTFT : X(\Omega) = \sum_{n=0}^{N-1} x[n] e^{-j\Omega n}$$





Original Signal

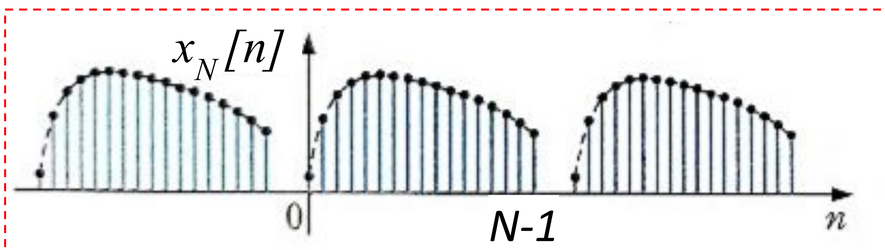
DTFT

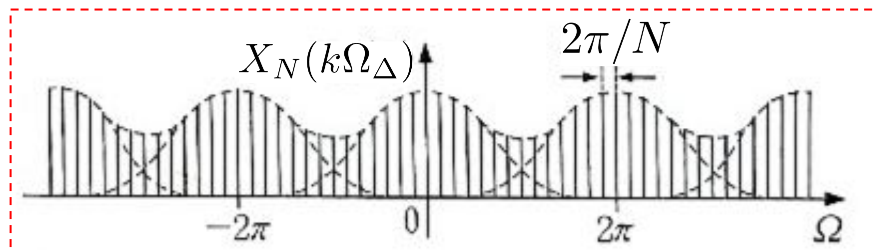
I-DTFT

Sampling:
N samples over
0<= Ω<=2π

$$\Omega_{\Delta} = \frac{2\pi}{N}$$

Repetition of Original Signal





Extract

I-DFT $: x[n] = \frac{1}{N} \sum_{n=0}^{N-1} X[k] e^{j\Omega_{\Delta} kn}$

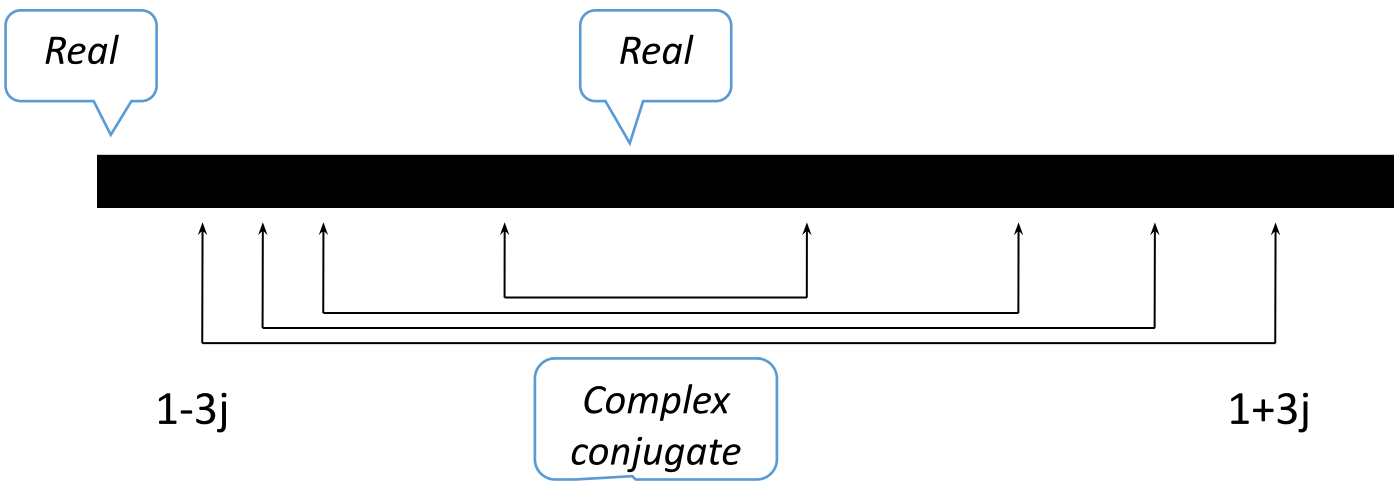
DFT $: X_N(k\Omega_{\Delta}) = \sum_{n=0}^{N-1} x[n] e^{-j\Omega_{\Delta} kn}$

DFT Properties

- Periodicity: $x[n] = x[N+n] \Leftrightarrow X[k] = X[N+k]$
- Symmetry if $x[n]$ is real: $X^{*}[k] = X[-k]$
- Convolution: $x[n]$ □ $y[n] \Leftrightarrow X[k]Y[k]$

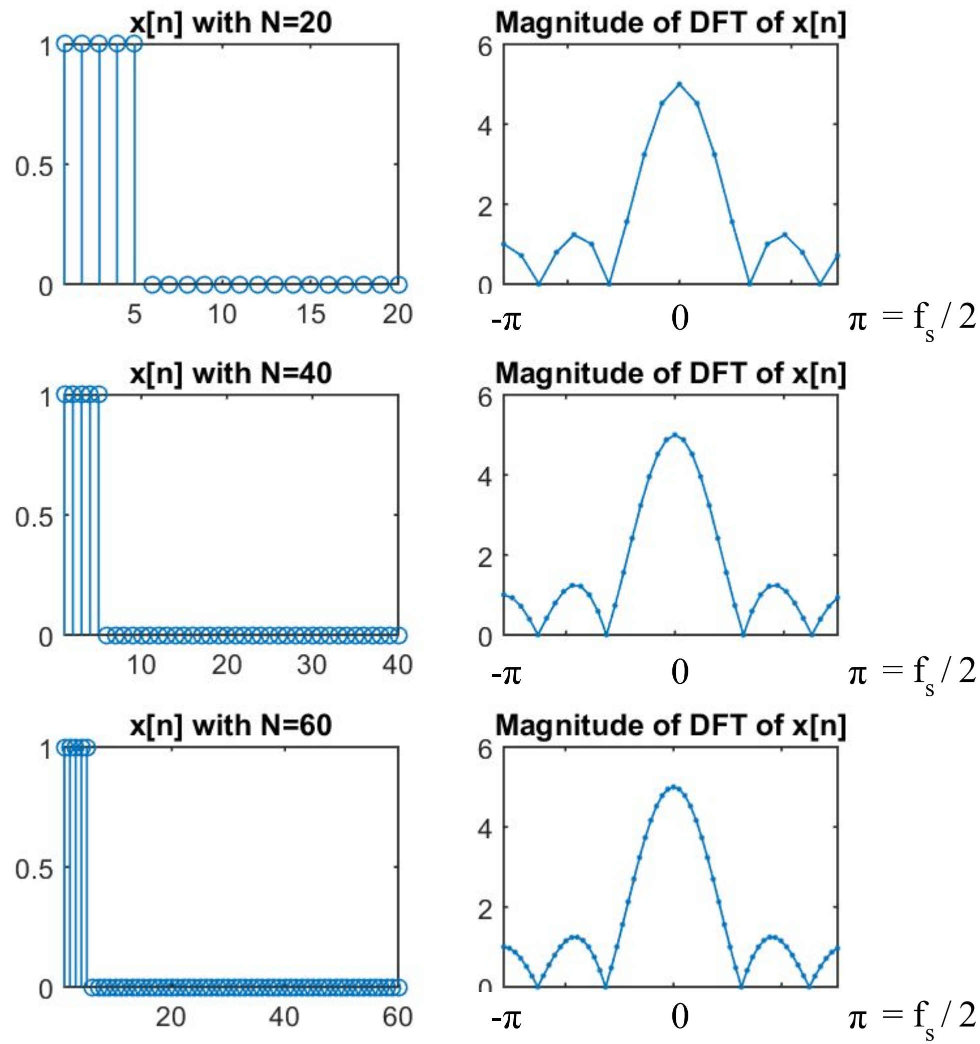
DFT Properties

- If x[n] is **real**, X[k] is conjugate symmetric



DFT Properties: Resolution

- For a given sampling frequency f_s, DFT resolution can be improved by increasing # samples
- Increasing # samples
  - Capture more samples from the original signal
  - Or, do **zero padding** – adding zeros at the end



http://mirlab.org/jang/books/audioSignalProcessing/ftDiscrete.asp?title=10-2%20Discrete%20Fourier%20Transform%20

DFT Example 1

- Find DFT of $x[n] = \{1, 2, 3, 4\}$

[redacted]

DFT Example 1

- Find DFT of x[n] = {1, 2, 3, 4}

[redacted]

$$e^{-j\frac{\pi}{2}} = \cos\frac{\pi}{2} - j\sin\frac{\pi}{2} = -j$$

[redacted]

$$= (1)(1) + (2)(1) + (3)(1) + 4(1) = 10$$

[redacted]

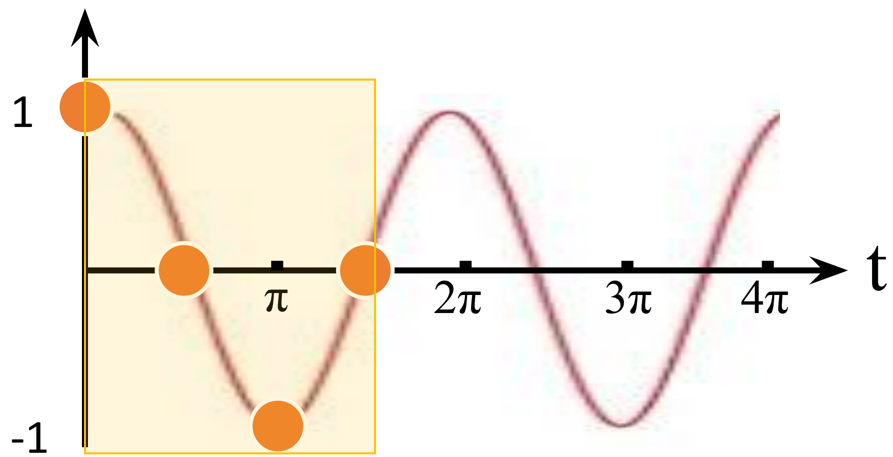
$$[10, -2+j2, -2, -2-j2]$$

DC=

$\sum$x[n]

DFT Example 1

- Find DFT of cos 2πt □ 1 hz signal
- Sampling frequency of 4 hz; i.e., fs = 4 hz
- Find DFT of x[n] = {1, 0, -1, 0}



[redacted]

Find DFT of x[n] = {1, 0, -1, 0}

[redacted]

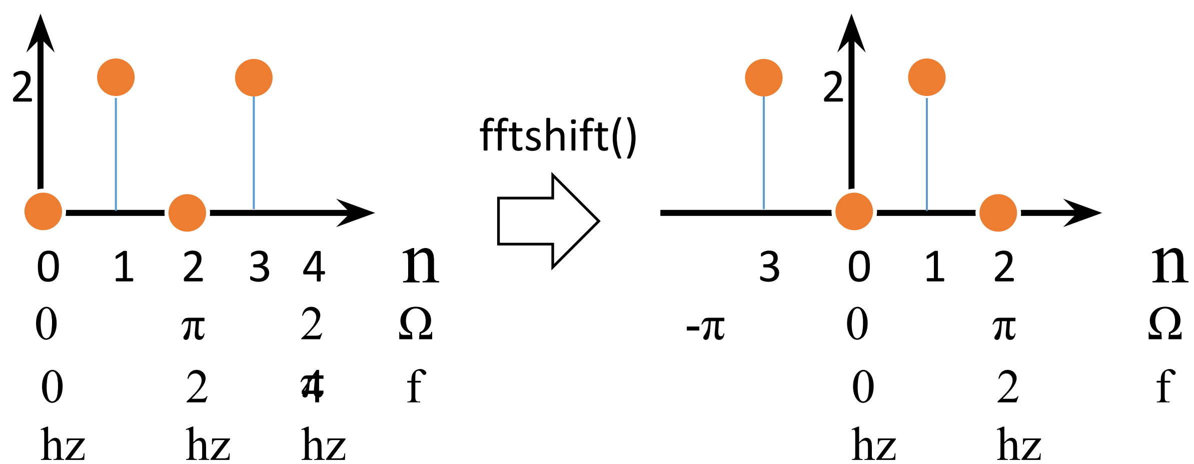
$$= (1)(1) + (0)(1) + (-1)(1) + (0)(1) = 0$$

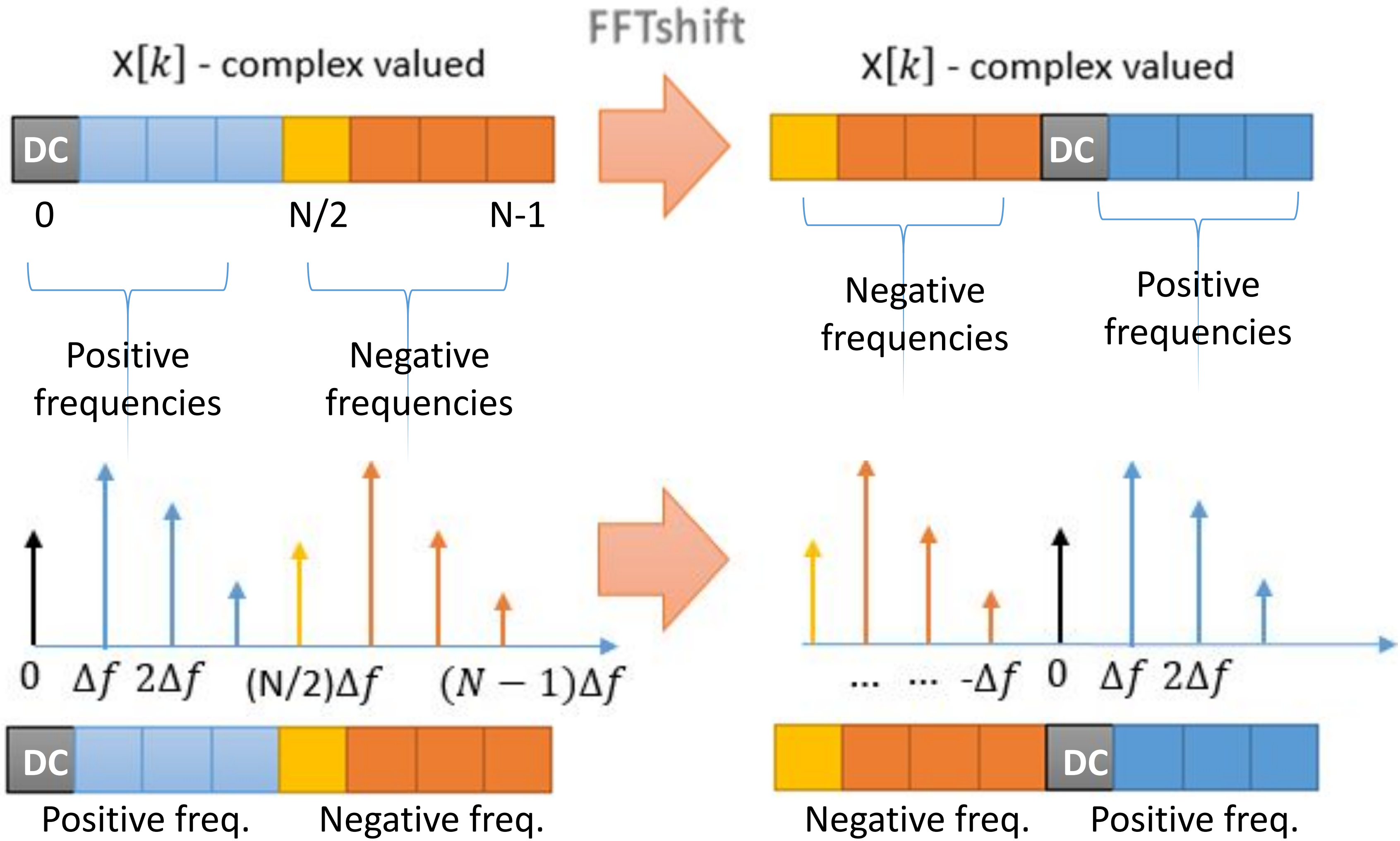
[redacted]

Δf = 4 hz / 4 = 1 hz

DC=$\sum$x[n]

$[0, 2, 0, 2]$



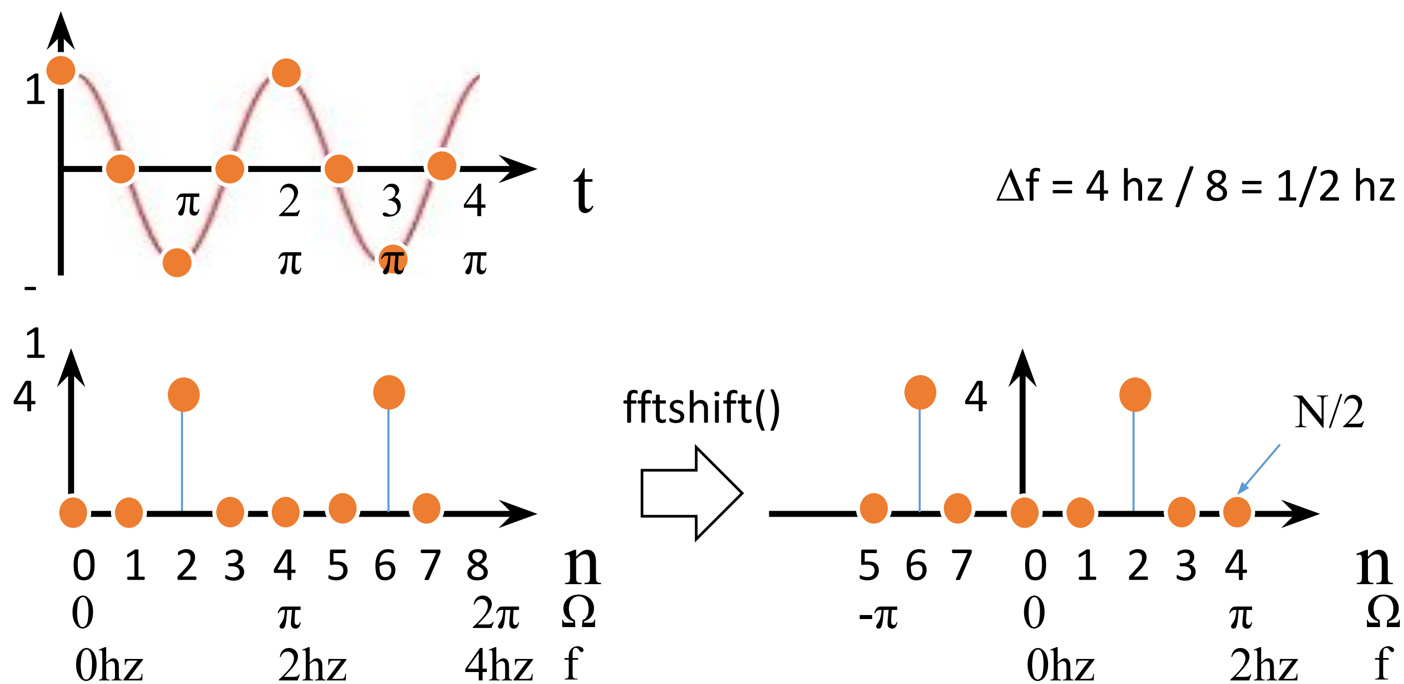


fftshift rearranges a multidimensional discrete Fourier transform, represented by a multidimensional array X, by shifting the zero-frequency component to the center of X

https://github.com/OrangeOwlSolutions/FFT/wiki/The-fftshift-in-CUDA

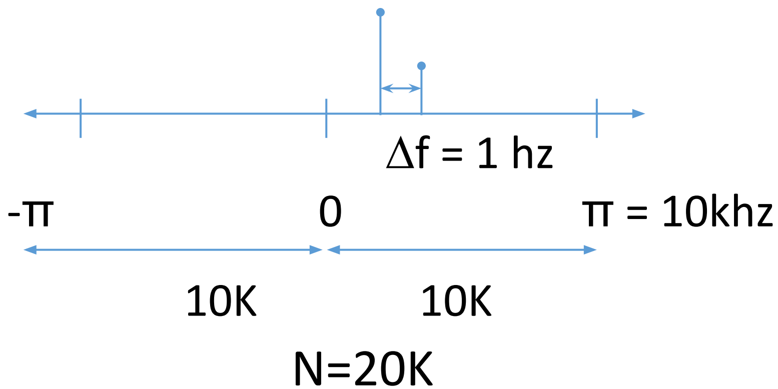
DFT Example 2

- Find DFT of cos 2πt □ 1 hz signal
- Sampling frequency of 4 hz; i.e., fs = 4 hz
- Find DFT of x[n] = {1, 0, -1, 0, 1, 0, -1, 0} (N=8)



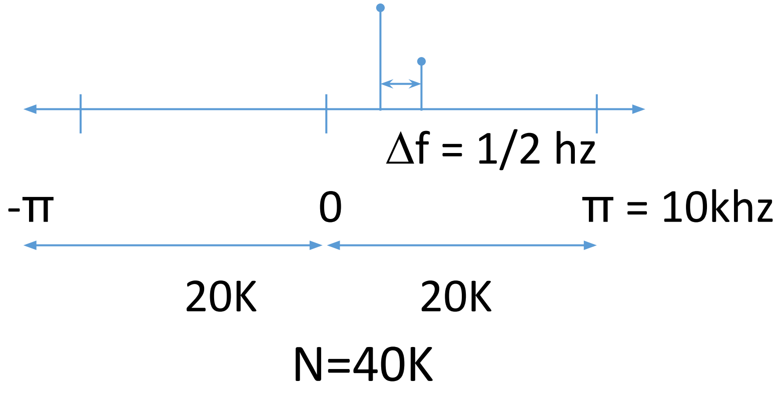
DFT: Sampling & Interpretation

- **<u>Signal's duration = 1s</u>**
- Signal's bandwidth f_b = 10khz (fixed)
- Nyquist sampling rate = $2 * f_b$ = **20khz**
- Total # of samples: N = 20khz * 1 s = **20k**
- Δf = 20k hz / 20k = 1 hz



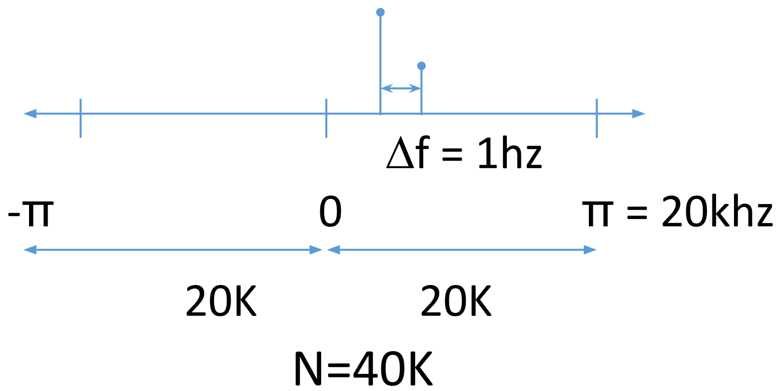
DFT: Sampling & Interpretation

- **Signal's duration = 2s (more samples from the original data)**
- Signal's bandwidth f_b = 10khz (fixed)
- Nyquist sampling rate = $2*f_b$ = **20khz**
- Total # of samples: N = 20khz * 2 s = **40k**
- Δf = 20k hz / 40k = 1/2 hz



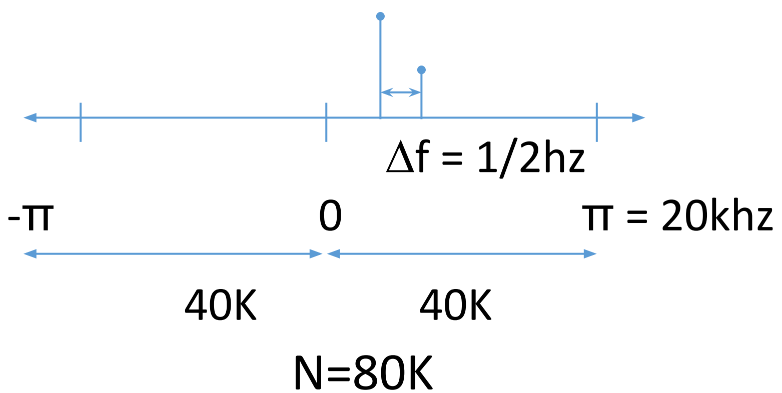
DFT: Sampling & Interpretation

- Signal's duration = 1s
- Signal's bandwidth f_b = 10khz (fixed)
- Nyquist sampling rate = $2*f_b$ = 20khz
- What happens if we increase the sampling rate?
  - $f_s > 2*f_b$ | for example, **f_s = 40 khz**
- Total # of samples: **N = 40khz * 1 s = 40k**
- Δf = 40k hz / 40k = 1 hz



DFT: Sampling & Interpretation

- Signal's duration = 2s (more samples from the original data)
- Signal's bandwidth f_b = 10khz (fixed)
- Nyquist sampling rate = $2*f_b$ = 20khz
- What happens if we increase the sampling rate?
  - $f_s > 2*f_b$ | for example, **f_s = 40 khz**
- Total # of samples: **N = 40khz * 2 s = 80k**
- Δf = 40k hz / 80k = 1/2 hz



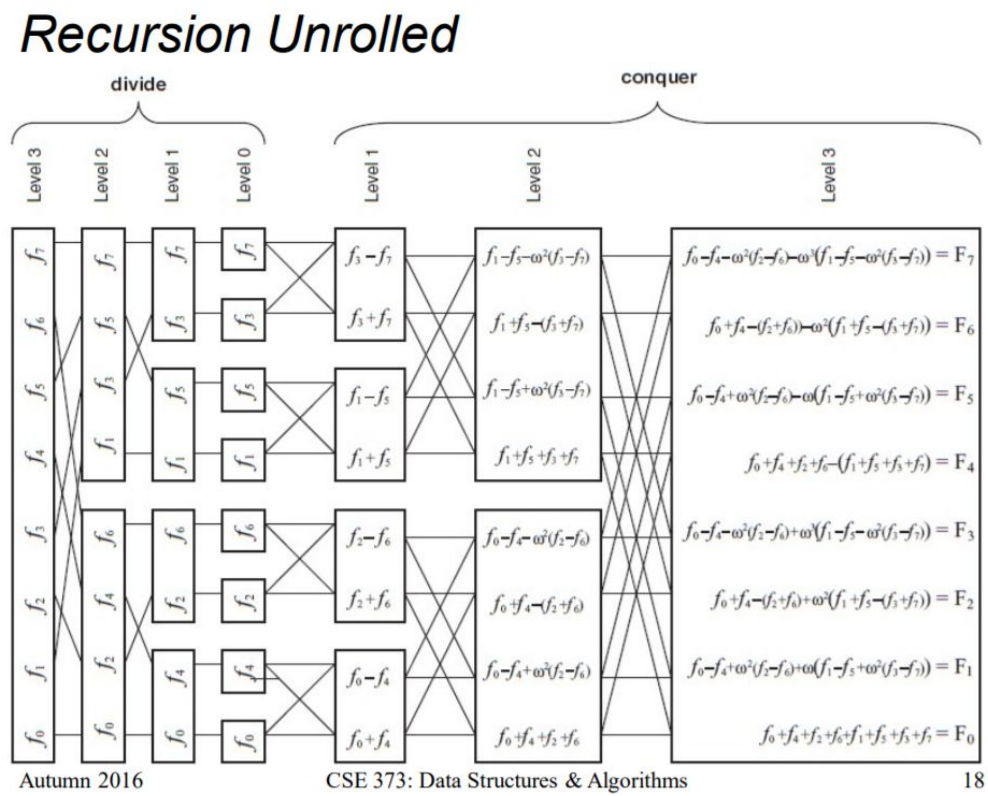
FFT (Fast Fourier Transform): $O(n^2)$ □ $O(n \log n)$

DFT as a linear transform

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

Recursive FFT

```
FFT(n, [a0, a1, …, an-1]):
  if n=1: return a0
  Feven = FFT(n/2, [a0, a2, …, an-2])
  Fodd  = FFT(n/2, [a1, a3, …, an-1])
  for k = 0 to n/2 – 1:
     ωk = e2πik/n
     yk     = Feven k + ωk Fodd k
     yk+n/2 = Feven k – ωk Fodd k
  return [y0, y1, …, yn-1]
```



https://courses.cs.washington.edu/courses/cse373/16au/slides/23-Divide-and-Conquer-the-FFT-6up.pdf

The `numpy.fft` Module

Function	***Purpose***	***Remarks***
`fft(`*s*`)`	Computes the forward DFT and returns the coefficients *F*	The returned array is a complex array.
`ifft(`*F*`)`	Computes the inverse DFT and returns the signal *s*	
`fftfreq(`*n*`,`*d*`)`	Computes the natural frequencies/wavenumbers. *d* is an optional sample spacing (default is 1).	The zero frequency is in the first position of the array, followed by the positive frequencies in ascending order, and then the negative frequencies in descending order
`fftshift(`*F*`)`	Shifts the zero frequency to the center of the array.	This can be used on either the frequencies or the spectral coefficients to put the zero frequency in the center.

http://snowball.millersville.edu/~adecaria/ESCI386P/esci386-lesson17-Fourier-Transforms.pdf

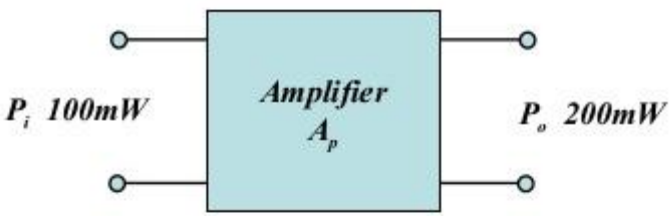
Energy vs. Power

Property	***Continuous***	***Discrete***
Energy	$E_x = \int_{-\infty}^{\infty} \lvert x(t)\rvert^2 \, dt < \infty$	$E_x = \sum_{n=-\infty}^{\infty} \lvert x[n]\rvert^2$
Power	$P_x = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} \lvert x(t)\rvert^2 dt < \infty$	$P_x = \lim_{N\to\infty} \frac{1}{2N} \sum_{n=-\infty}^{\infty} \lvert x[n]\rvert^2 < \infty$

Energy: joule
Power: joule per second

Power Gain (or Power Ratio)

Decibel - Power Gain



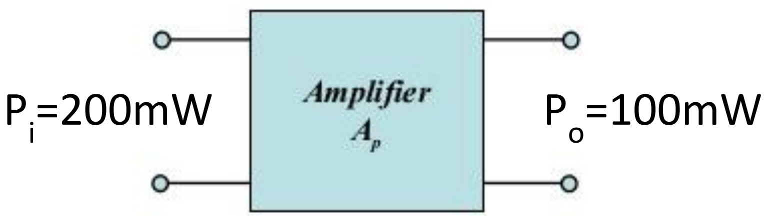
$$A_p = \frac{P_o}{P_i} = \frac{200}{100} = 2$$

$$A_p = 10\log_{10}\frac{200}{100} = 3dB$$

(log 2 = 0.301)

A 3dB gain is a doubling of power

Decibel - Power Gain



$$A_p = \frac{P_o}{P_i} = \frac{100}{200} = 1/2$$

$$A_p = 10\log_{10}\frac{100}{200} = -3dB$$

A −3dB gain is a halving of power

Decibel (dB)

Two levels of power can be compared using a unit of measure called the ***bel***.

$$B = \log_{10} \frac{P_2}{P_1}$$

The ***decibel*** is defined as:

1 bel = 10 decibels (dB)

https://www.slideshare.net/touqeerjumani/active-filters-43112925

	Power ratio	Voltage ratio
-20 dB	0.01	0.1
-10 dB	0.1	0.32
-3 dB	0.50	0.71
-1 dB	0.74	0.89
0 dB	1	1
1 dB	1.26	1.12
3 dB	2.00	1.41
10 dB	10	3.16
20 dB	100	10
$n \cdot 10$ dB	10^n	$10^{n/2}$

Power Ratio	Voltage Ratio	Decibel Value
1	1	0 dB
2	1.4	3 dB
4	2	6 dB
10	3.16	10 dB
100	10	20 dB
1,000	31.6	30 dB
10,000	100	40 dB
100,000	316	50 dB
1,000,000	1,000	60 dB
10,000,000	10,000	80 dB
100,000,000	100,000	100 dB

$$\text{Power: } \mathrm{dB} = 10\log_{10}\frac{P_2}{P_1}$$

$$\text{Amplitude: } \mathrm{dB} = 20\log_{10}\frac{A_2}{A_1}$$

(Voltage)

https://www.researchgate.net/figure/Overview-of-dB-key-values-and-their-conversion-into-power-and-voltage-ratios_tbl1_51989700

DSP Basics

- ADC & DAC
- Signal Basics (Sine Wave)
- ADC: Sampling - What's proper sampling rate?
  - Nyquist sampling rate
  - Aliasing
- Discrete Fourier Transform (DFT)
  - How to do?
  - DFT and its relationship with other transforms
  - DFT properties: symmetry, resolution
  - DFT examples
  - Using DFT in Python
- Energy vs. Power