Lecture 18

Modern Topics: Spectrum Analyzer, FFT, and OFDM

I. Spectrum Analyzer

II. DFT and FFT

III. Digital Communication and OFDM

ak

CTFS DTFS

CTFT DTFT

I. Spectrum Analyzer

- In this section, we further illustrate the concept of sampling and the relationship among different variants of Fourier analysis by considering how a modern digital spectrum analyzer operates.

important!

- Spectrum analyzers (SA) is a piece of equipment for analyzing the spectrum of signals. It can be quite expensive. One that can handle frequency up to 8 GHz, as shown below, cost tens of thousands of US$$$

analysis the eletrospectrum!

Electrical Signal Spectrum Analyzer (Tektronix)

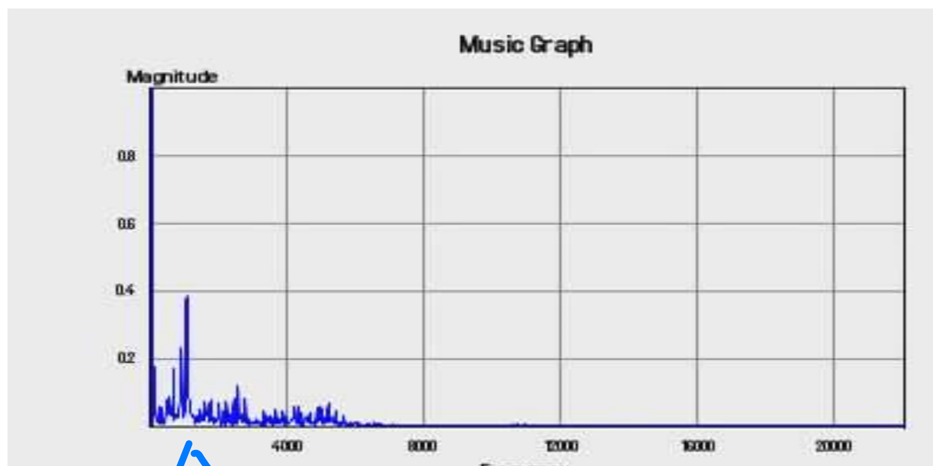
Frequency range: DC to 8 GHz

http://www.testequipmentconnection.com/manufacturer/Tektronix/191/198

Demo – Spectrum Analyzer In Action

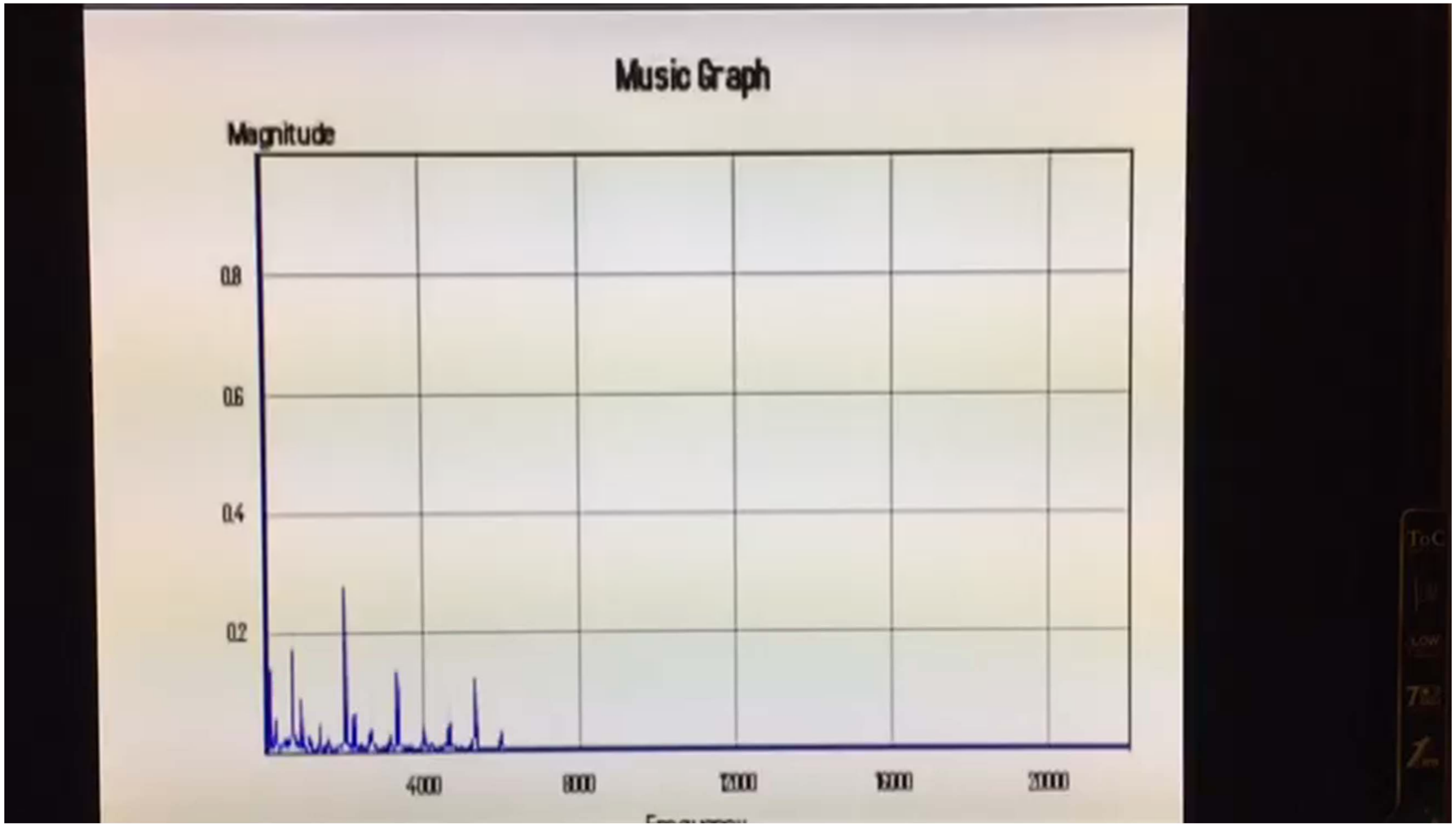
- The next screen shows a spectrum analyzer (SA) in action analyzing an audio music signal, showing the spectrum in the frequency range of 0 to 22,000 Hz

  http://en.wikipedia.org/wiki/File:My_Songo_Real_Time_Analysis.ogg



隨时间改變

Demo – Spectrum Analyzer In Action



$CTFT = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$

SA Operation (I) – Taking a finite chunk of signal

- Can the SA compute the FT integral?

not practical !!!

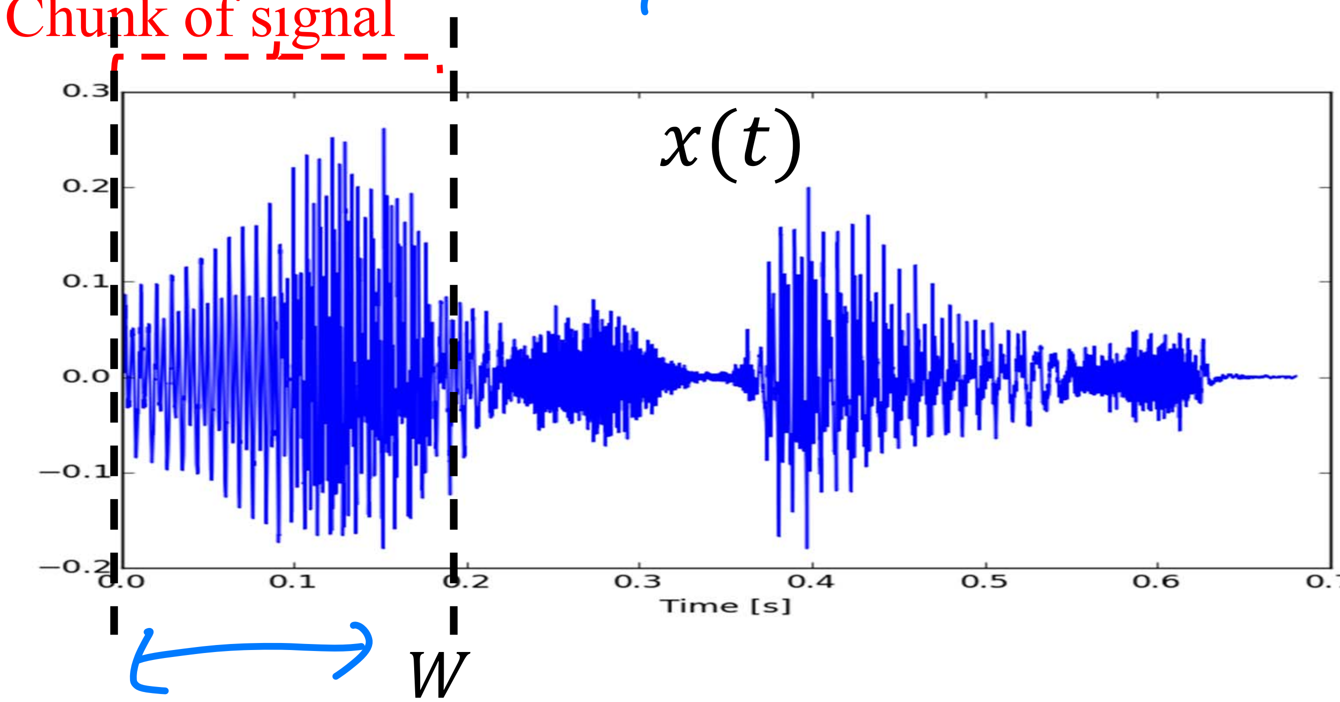
$$X(j\omega) = \int_{-\infty}^{\infty} x(t)\, e^{-j\omega t}\, dt$$

Cannot wait till $t = \infty$ to compute this integral!

- The SA has to analyze one "chunk" of the signal at a time. ***That's why the spectrum in the demo changes with time***. Assume each chunk has duration W and let's call it $x_W(t)$.

$x_W(t)$
A finite duration
Chunk of signal

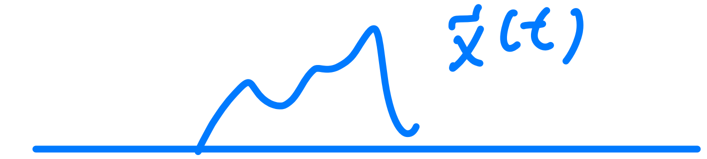
$\sim 4W!$



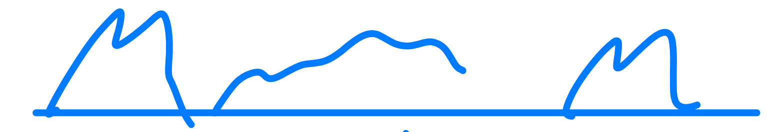
⇒ ***How should we compute its spectrum?***

use sliding windows!

AKW



How to Compute Spectrum of a Finite Duration Chunk



We can compute its FT: $$X_W(j\omega) = \int_{-\infty}^{\infty} x_W(t)\, e^{-j\omega t}\, dt \;=\; \int_0^W x_W(t)\, e^{-j\omega t}\, dt$$

ak is passion sum periodic!

Or we can compute its FS: $$a_k = \frac{1}{W}\int_0^W x_W(t)\, e^{-jk\frac{2\pi}{W}t}\, dt = \frac{1}{W} X_W\!\left(k\,\frac{2\pi}{W}\right)$$

$a_k = \frac{1}{W} X\left(jk\frac{2\pi}{T}\right)$

normalization

smell like digital device, treat it as periodic!

frequency

Is there any difference ?

ak is sample value of the spectrum

- We observe: $T \to W$

1. The FS coefficients are sampled values of the FT (with scaling of $1/W$)

   finite duration signal ← Bandlimited!

2. Either the FT or FS fully specifies $x_W(t)$ - we can synthesize $x_W(t)$ from either one. So for a finite duration signal, ***the sampled values of the FT fully specify the FT***! This is analogous to sampling theorem!

Duration of Chunks and Frequency Resolution

what should be W is?

fundamental frequency!

- How large a chunk of signal should the SA analyze at a time?

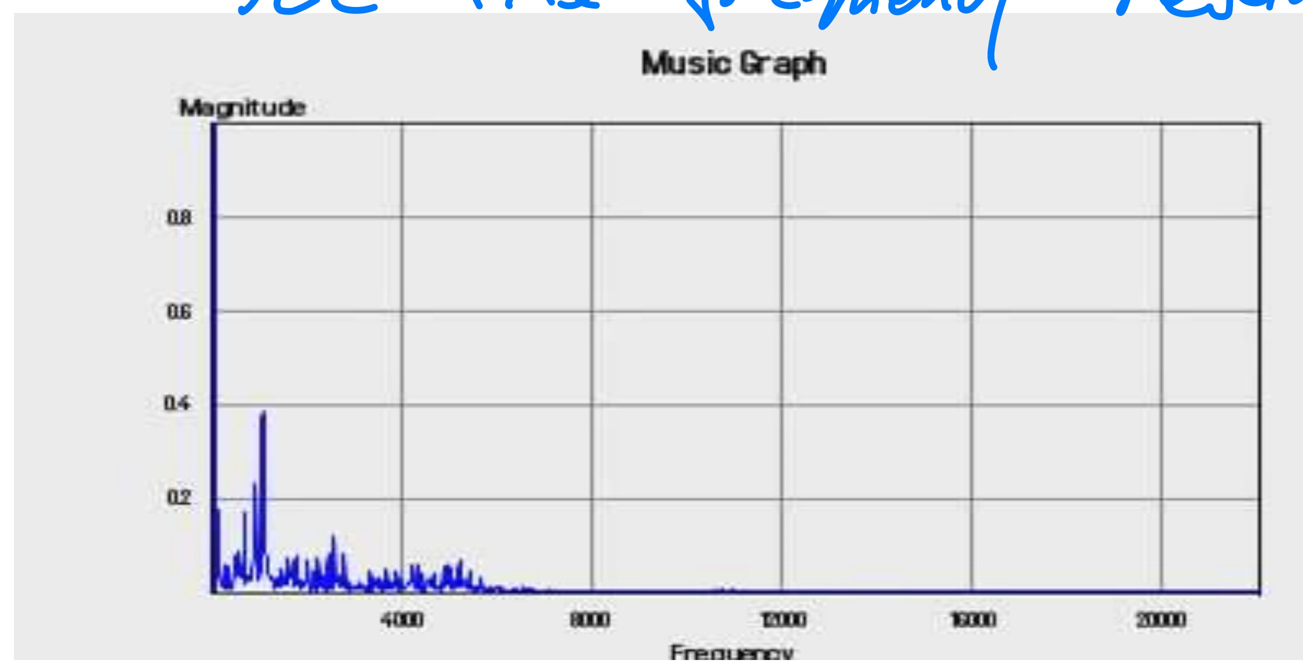
Note that $1/W$ Hz ($2\pi/W$ rad/s) is the fundamental frequency for the FS and thus the frequency resolution of the spectrum shown. Say, if we want to resolve the spectrum down to 10 Hz, the duration of the chunk needs to be larger than 0.1 second.

want see fine frequency resolution

longer you need to wait!

To see if there is some low frequency components in the signal, or to have fine frequency resolution, we must wait for a longer time!

A resolution of 10 Hz here is beyond what our eyes can see



不知 ak, X(jw) ???

ND = W

$\tilde{a}_k$: k from 0 to N-1!

freq of k-th frequency $\Omega_k = k \cdot \frac{2\pi}{N}$ (discrete time frequency!)

highest frequency $= \pi$ $= \frac{N}{2}$

SA Operation (II) – Sampling and Numerical Integration

- Instead of doing integration, the SA, as a digital device, samples the integrand at interval Δ, and performs a numerical integration (i.e., summation) which is again a staircase approximation of the integral:

$\frac{1}{N} = \frac{\Delta}{W}$

FS Analysis Integral: $$a_k = \frac{1}{W}\int_0^W x_W(t)\, e^{-jk\frac{2\pi}{W}t}\, dt$$

sample!

Number of samples $N = \frac{W}{\Delta} \Rightarrow \frac{\Delta}{W} = \frac{1}{N}$

Numerical Integral/ Staircase approximation: $$\tilde{a}_k = \frac{1}{W}\sum_{n=0}^{N-1} x_W(n\Delta)e^{-jk\frac{2\pi}{W}n\Delta}\,\Delta = \frac{1}{N}\sum_{n=0}^{N-1} x_W(n\Delta)e^{-jk\frac{2\pi}{N}n}$$

summation!

$x[n]$

N from $-\frac{N}{2}+1$ to $\frac{N}{2}$ (even; highest frequency!)

This is a DTFS analysis sum!

The numerical integral ***is a DTFS***!

一模一样

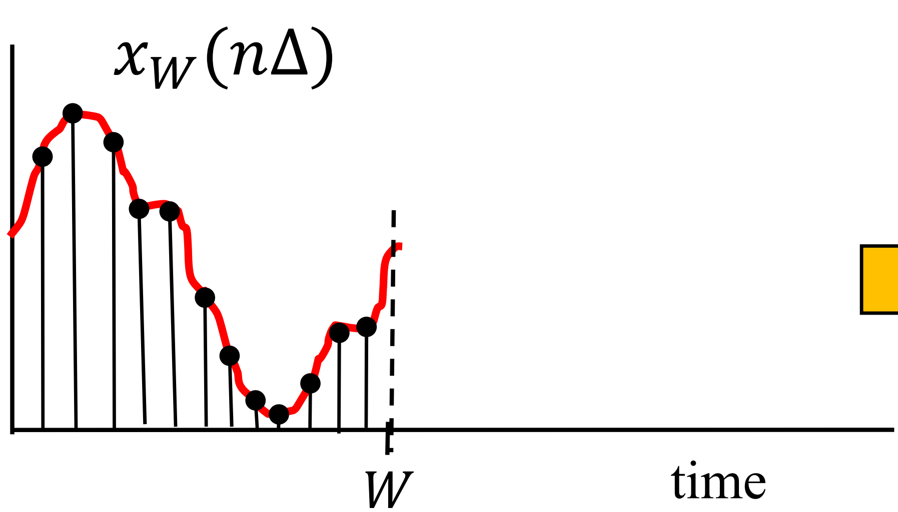
$\Omega = \pi$

$\omega = \frac{\pi}{\Delta}$

$f = \frac{1}{2\Delta} = \frac{f_s}{2}$

$\frac{1}{\Delta}$ is sampling frequency!

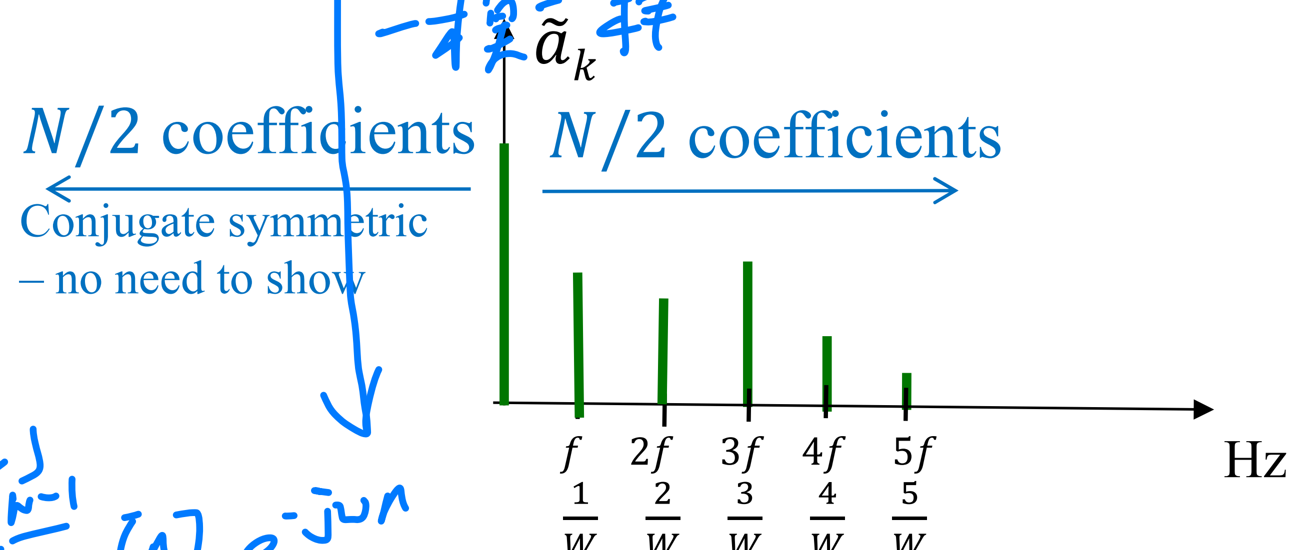
for continuous signal, k can be very large!



N sample values of $x_W(t)$

DTFS

$a[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\, e^{-j\omega n}$



N DTFS coefficients

$k = \frac{N}{2}$

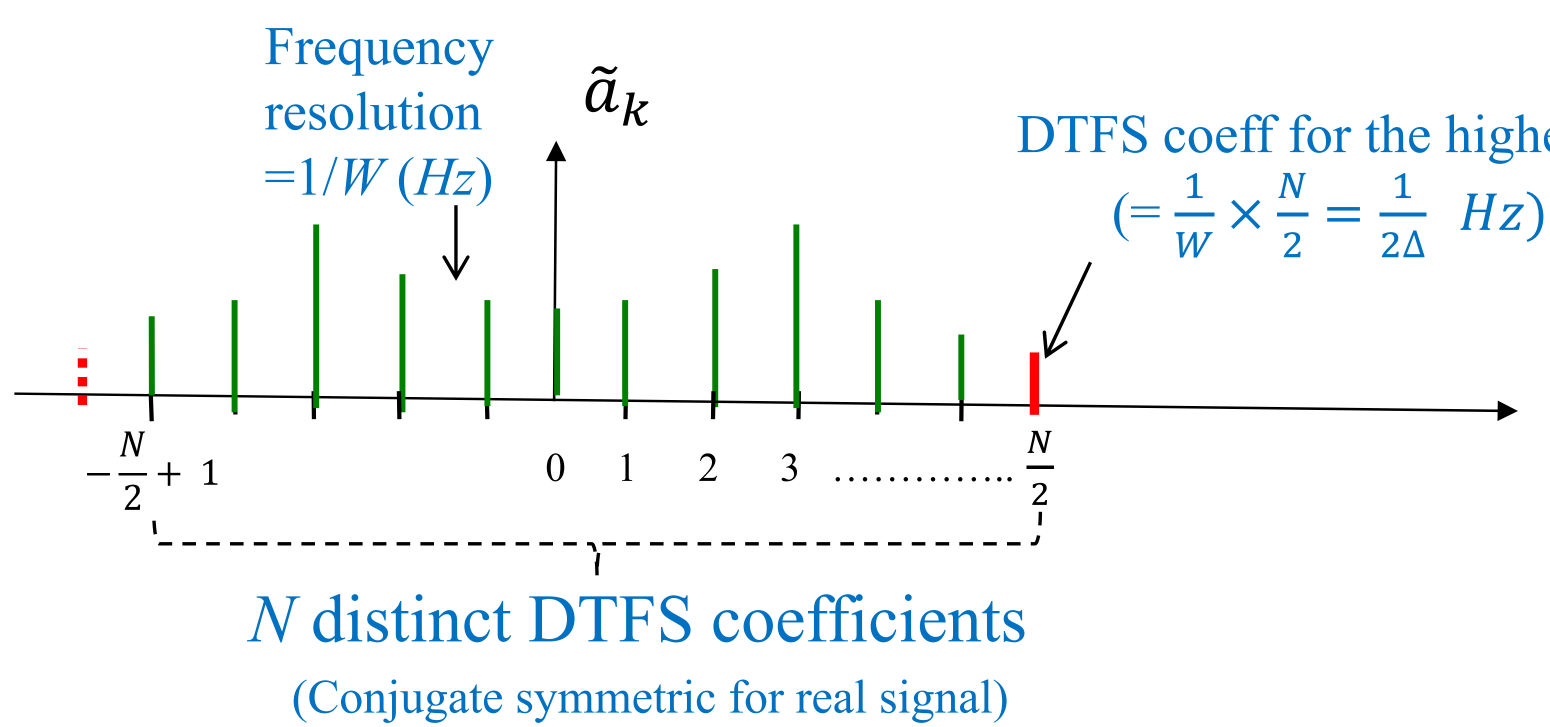
Choice of the Sampling Interval for Numerical Integration

- The numerical integral is the **DTFS of the sampled sequence**: $x_W[n] = x_W(n\Delta)$

$$\tilde{a}_k = \frac{1}{N}\sum_{n=0}^{N-1} x_W(n\Delta)e^{-jk\frac{2\pi}{N}n} \quad ; \quad x_W[n] = x_W(n\Delta)$$

- For a DTFS, the FS is periodic; i.e., $\tilde{a}_{k+N}=\tilde{a}_k$. Or viewed alternatively, there are only $N = W/\Delta$ distinct $\tilde{a}_k$, with $k = N/2$ (assume N even) representing the highest frequency harmonic at frequency of $1/2\Delta$.

$k = \frac{N}{2}$



- Thus, for the SA to analyze signals with frequency as high as 8 GHz, the SA must sample signals at 16 GHz! This is consistent with our understanding that the highest frequency we can see in a DT signal is ½ the sampling frequency.

- In summary, when doing spectrum analysis,

Want fine frequency resolution ⟺ Use long chunk of signal large W

Want to handle high frequency ⟺ Need high sampling rate small Δ

$$\textbf{Frequency resolution} = \frac{1}{W}\ \text{Hz} = \text{reciprocal of duration of chunk}$$

$$\textbf{Highest frequency} = \frac{1}{2\Delta} = \frac{f_s}{2}\ \text{Hz} = \text{½ of sampling rate}$$

Lecture 18

Chapter 7: Spectrum Analyzer, FFT, and OFDM

I. Spectrum Analyzer and Spectrogram

II. DFT and FFT

III. Digital Communication and OFDM

II. Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT)

- Instead of computing DTFS, modern digital signal processing is actually based on what is called the ***Discrete Fourier Transform* (DFT)**.
- ***DFT is the same as DTFS***, except for the absence of the scaling factor. The difference is only a matter of conventional preference.

$$X[k] = \sum_{n=0}^{N-1} x(n) W_N^{kn}$$

$$W_N = e^{-j\frac{2\pi}{N}}$$

DTFS

$$a_k = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-jk\frac{2\pi}{N}n} \qquad x[n] = \sum_{k=0}^{N-1} a_k e^{jk\frac{2\pi}{N}n}$$

DFT

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-jk\frac{2\pi}{N}n} \qquad x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{jk\frac{2\pi}{N}n}$$

$$X[k] = a_k N$$

no normalization!

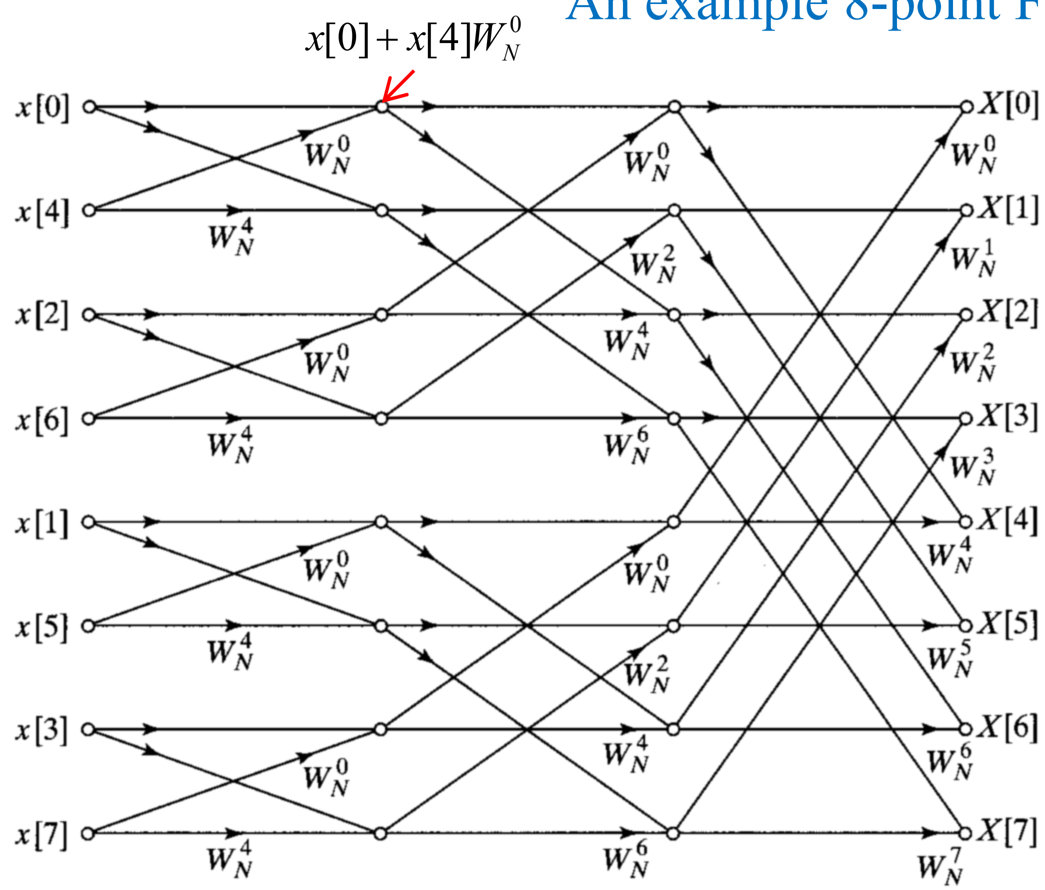
DFT and DTFS are the same except for where you put the 1/N!

huge N, will have great advantage!!!

Fast Fourier Transform (FFT)

- ***FFT*** is an algorithm discovered in 1965 for the efficient calculation of DFT or DTFS.
- For a DT sequence of duration N, FFT takes order of $O(Nlog_2N)$ multiplications to compute the entire set of $X[k]$ (or DTFS coefficients a_k).
- FFT is based on $\log_2 N$ stages of computation of intermediate values as illustrated by the computation network below:

An example 8-point FFT; N=8



$$\boxed{W_N^k = e^{-j2\pi k/N}}$$

Application of FFT

- In 4G mobile, WiFi, cable modem, and new coherent high-speed optical communication systems, information bits are carried by the Fourier coefficients in the signal. FFT enables us to compute thousands of these coefficients efficiently.
- FFT also allows us to perform convolution much faster by processing the input signal "chunk by chunk" (future topic in ELEC3100)
  - Finite Impulse Response (FIR) filter $h[n]$ with duration N_1
  - Choose an FFT size N that is power of 2 ($n = 1024, 2048, \ldots 65536, etc$)
  - Chop signal $x[n]$ into chunks $x_{chunk\ i}[n]$ with N_2 samples in each chunk such that $N_1 + N_2 - 1 \le N$
  - N-point FFT of $h[n]$ computed once to obtain frequency response $H(k)$
  - N-point FFT of each chunk $x_{chunk\ i}[n]$ to obtain its DFT $X_{chunk\ i}[k]$ – $O(Nlog_2N)$ multiplications
  - Multiply $H(k)$ with $X_{chunk\ i}[k]$ – N multiplications
  - Inverse FFT of the result to obtain $y_{chunk\ i}[n] = x_{chunk\ i}[n] \circledast h[n]$ - $O(Nlog_2N)$ multiplications
    Circular convolution and regular (linear) convolution are equivalent because $N_1 + N_2 - 1 \le N$
  - Overlap and add the individual output chunks $y_{chunk\ i}[n]$ to obtain the output $y[n]$
- GPS and audio and image/video processing systems also use FFT extensively.
- FFT is one of the most fundamental algorithms driving the modern world.

Lecture 18

Chapter 7: Spectrum Analyzer, FFT, and OFDM

I. Spectrum Analyzer and Spectrogram

II. DFT and FFT

III. Digital Communication and OFDM

$$y(t) = \sum_{m=0}^{M} \underbrace{\rho_m}_{\text{attenuation}} x(t - \underbrace{\tau_m}_{\text{delay}})$$

Want $T >> \Delta$
$\Rightarrow$ low baudrate! cannot send many bits per second $\Rightarrow$ overload each symbol!

# of symbols per second != baud rate!

III. Digital Communications and OFDM

represent the i:th symbol'.

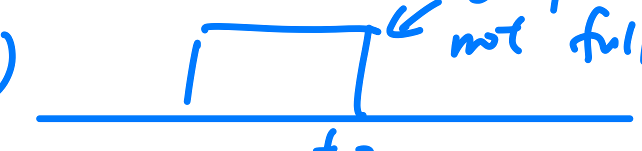
- Digital communications is to send sequence of 0 and 1 bits.
- For example, digital TV, which first appeared in 1994, was launched in Hong Kong in 2007. In digital TV standards, the video signal is encoded into a multi-megabit per second bit stream.

(must > max.delay!) $t_0 > \Delta$, otherwise have ISI!

- Recall we introduce a simple data stream signal $x(t) = \sum_i \sigma_i r(t - iT)$ in Lecture 2, Section 4, where:

  T is the symbol duration,

  window!

  $r(t)$  only t_0, not full period! t_0

  $$r(t - iT) = \begin{cases} 1 & iT \le t < (i+1)T \\ 0 & otherwise \end{cases}$$ is a window covering the i-th symbol,

  pulse  $\times\ \sigma_i = \begin{cases} 0, A \\ -A, A \\ -A, -\frac{1}{3}A, \frac{1}{3}A, A \end{cases}$ 2 bits!

  and σ_i is some signal amplitude value conveying 0 and 1 bits.

- In radio and optical communications, the data stream signal $x(t)$ is not transmitted as is in baseband. Instead, it is used to modulate, or vary, a sinusoidal electro-magnetic (EM) wave carrier. We mentioned previously that we can vary the *amplitude*, *phase*, or *frequency* of the EM carrier to convey bits.
- In this section, we will show that we can easily represent amplitude and phase modulation by allowing σ_i to be complex.

(low symbol rate!)
4G, 13000 symbols per second
$\Downarrow$
$T = \frac{1}{13000} = 76\mu s/s$

15 kHz, is fundamental freq. f_0 of complex sinusoid, $T_0 = \frac{1}{15kHz}$

I/Q Channel

- In radio/optical communication, we can actually use two carrier waves at the same frequency but 90^{o} out of phase (cosine and sine waves) to transmit two information signals at the same time.
- By convention, the carrier lagging in phase by 90^{o} is called the I (In-Phase) channel, and the carrier leading in phase by 90^{o} is called the Q (Quadrature Phase) channel.

For convenience, I simply use ω instead of ω_0

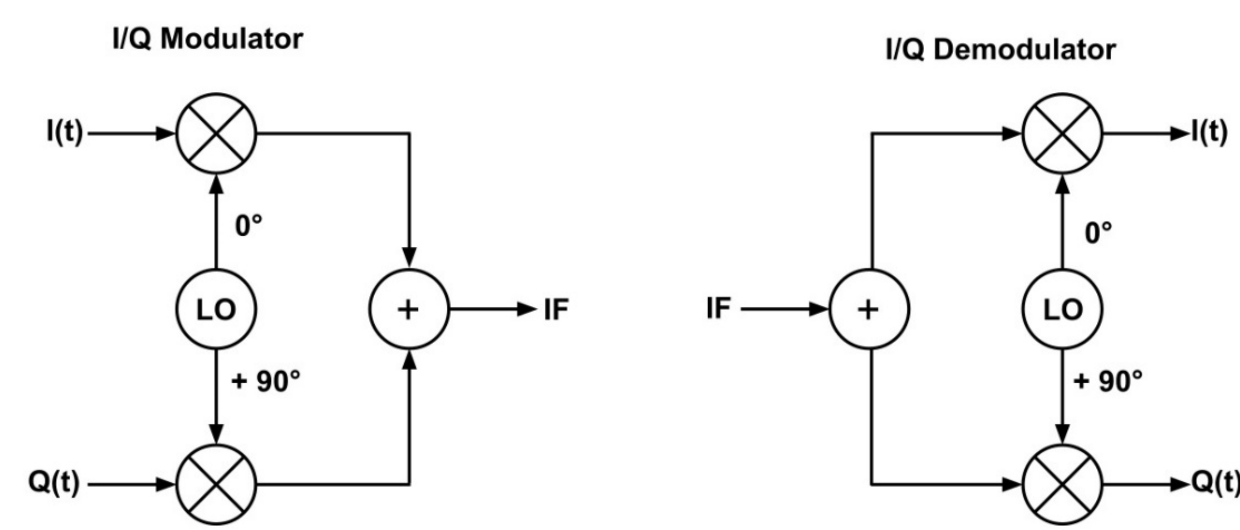
I/Q Transmitter

$$x_I(t)\cos(\omega t) + x_Q(t)\cos\left(\omega t + \frac{\pi}{2}\right)$$

or $x_I(t)\cos(\omega t) - x_Q(t)\sin(\omega t)$

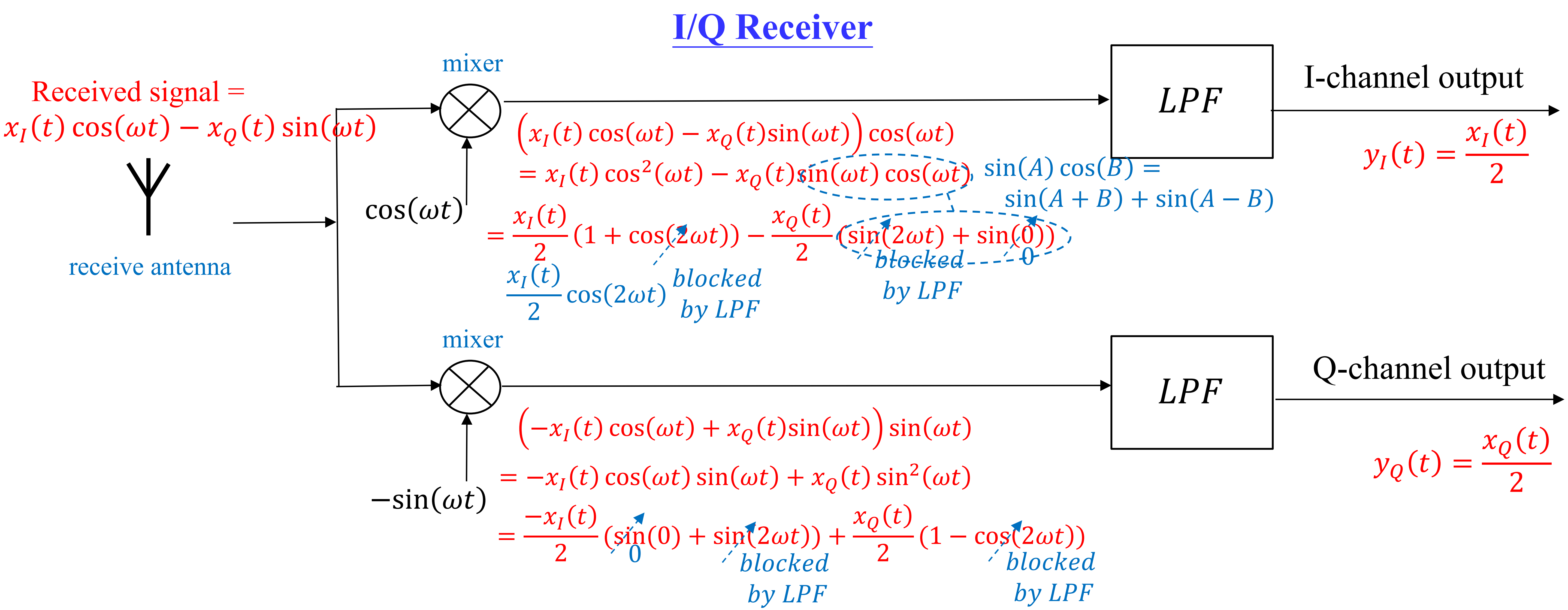
$x_I(t), x_Q(t)$, are two information signals.

We can think of them as one complex-valued information signal, with $x_I(t)$ being the real part of the signal and $x_Q(t)$ being the imaginary part.



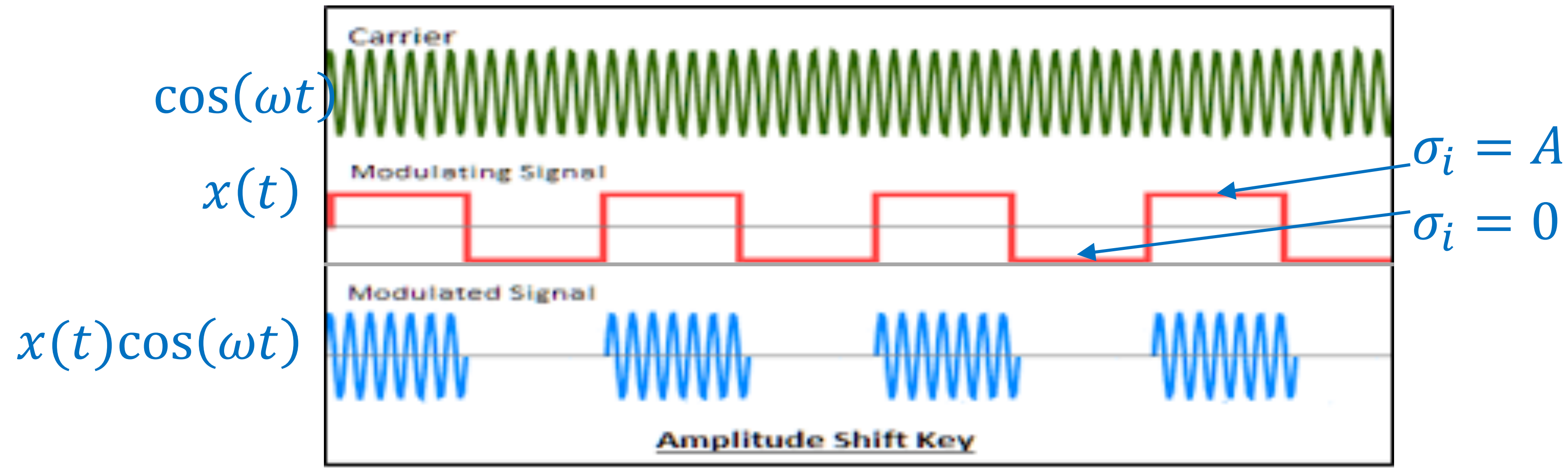
I/Q Channel Receiver

- At the receiver, we mix the received signal separately with the in-phase carrier and quadrature carrier to recover the I-channel and Q-channel information signals:

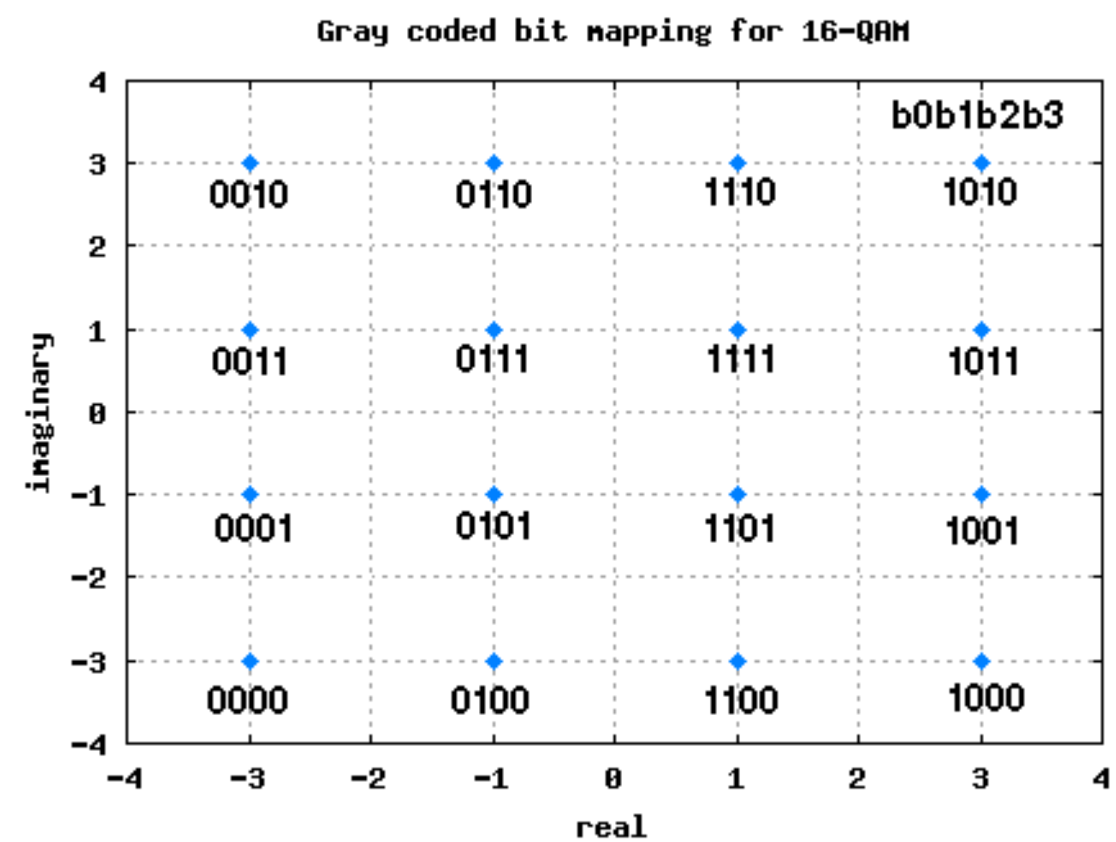


ASK and QAM

- In *on-off keying*, which is the simplest form of *Amplitude Shift Keying* (ASK) we send a pulse for a “1” bit and no pulse for a “0” bit; i.e., the information signal $x(t)$ is a sequence of on-off pulses and $\sigma_i = \{0, A\}$; σ_i is real and equals either 0 or A.

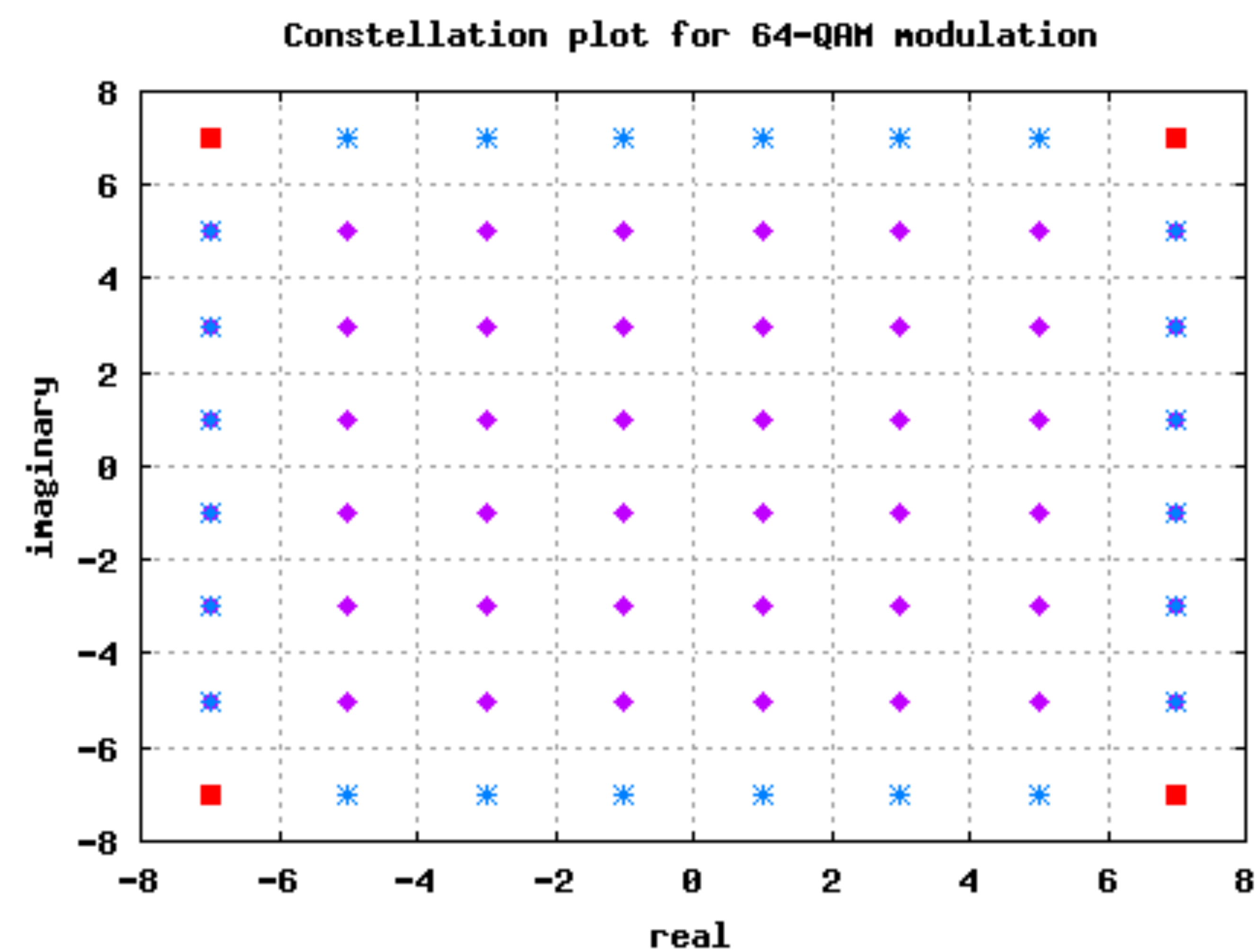


- In *Quadrature Amplitude Modulation* (QAM), σ_i is viewed as complex. In 16-QAM, $\sigma_i = \{\alpha_i + j\beta_i;\, \alpha_i, \beta_i = -3, -1, 1, 3\}$ - the real and imaginary parts of σ_i may each take one of four possible values so σ_i has 16 possible values and indicates 4 bits.



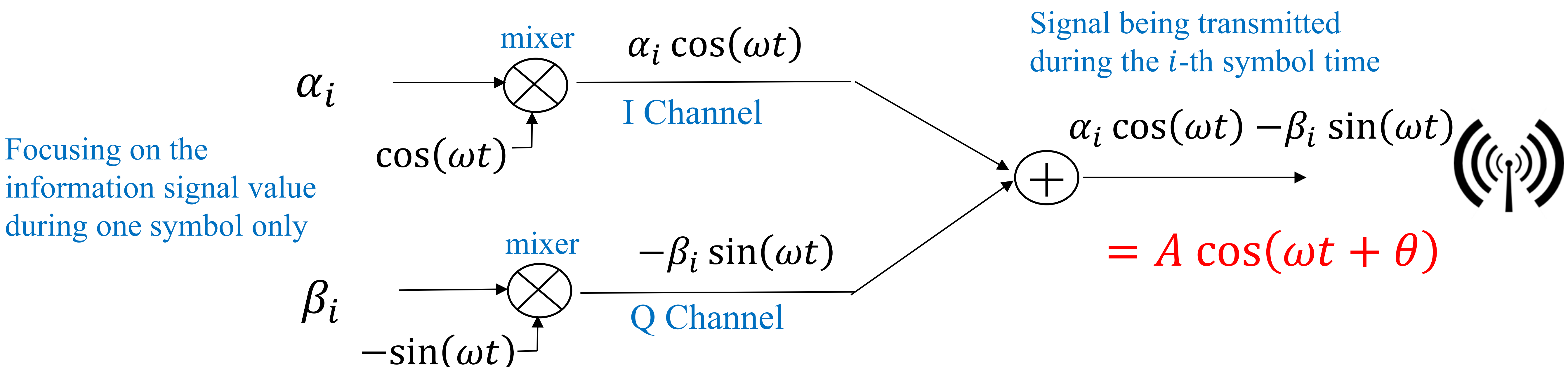
Signal Constellation

- In 64 QAM, σ_i takes one of 64 values to indicate 6 bits.
- So why don't we use 128-QAM, or 256-QAM, to transmit more bits per symbol?
- The answer is that communication channels invariably add noise. If we use too many possible signal values without increasing the power in the signal, the bit error rate will increase.
- The map indicating the possible symbol values on the complex plane is called a signal *constellation*.
- Next, we show that QAM is in effect an amplitude plus phase modulation.



I/Q Transmitter for 16 QAM

- So what does a complex σ_i mean?
- During each symbol time, to transmit σ_i, we modulate the in−phase carrier by $\alpha_i = Re\{\sigma_i\}$ and the quadrature carrier by $\beta_i = Im\{\sigma_i\}$. Then we transmit the sum:



$\alpha_i = Re\{\sigma_i\} = A\cos(\theta)$

- Signal transmitted for the i-th symbol is $\alpha_i \cos(\omega t) - \beta_i \sin(\omega t)$.

  We can re-express it as $A\cos(\theta)\cos(\omega t) - A\sin(\theta)\sin(\omega t)$ where A and θ are the magnitude of phase of the symbol σ_i.
- Using the trigonometric identity $\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$, we see that the transmit signal is simply $A\cos(\omega t + \theta)$! The transmitted signal is simply one wave with an amplitude and phase as specified by σ_i!

Complex Representation of Modulated Signal

- The transmitted signal for the i-th symbol is $\alpha_i \cos(\omega t) - \beta_i \sin(\omega t) = A \cos(\omega t + \theta)$.
- A more concise representation of the transmitted signal is $\sigma_i e^{j\omega t}$.

We can think of the transmitted RF signal $A \cos(\omega t + \theta)$ as the real part of $\sigma_i e^{j\omega t}$,

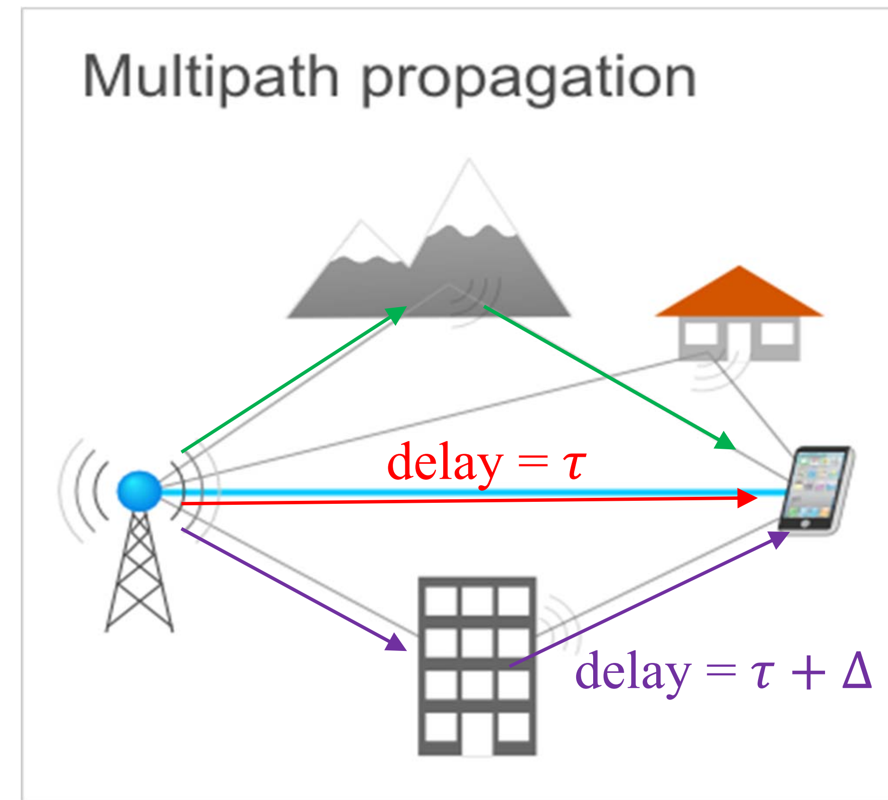
since $A \cos(\omega t + \theta) = \mathrm{Re}\{\sigma_i e^{j\omega t}\}$.

- Over all symbols, the transmitted RF signal can be presented mathematically by:

$$\text{transmitted RF signal} = \underbrace{\sum_i \sigma_i r(t - iT)}_{\substack{x(t) \\ \text{The information signal} \\ \text{that convey bits}}} \underbrace{e^{j\omega t}}_{\substack{\text{The complex sinusoid} \\ \text{representing the} \\ \text{I- and Q- carriers}}}$$

Orthogonal Frequency Division Multiplexing (OFDM)

- To achieve high bit throughput, we need to transmit symbols at a high rate. Want $T \to 0$
- But in mobile communications, multipath spreading prevents us from transmitting symbols at high rates. If the largest delay difference is Δ, the duration of each symbol must be much greater than Δ.
- OFDM is the technique used in WiFi, 4G, and DSL communications. The idea is to use a low symbol rate but a very large number of subcarriers closely spaced in frequency to transmit many bits per symbol.



Orthogonal Frequency Division Multiplexing (OFDM)

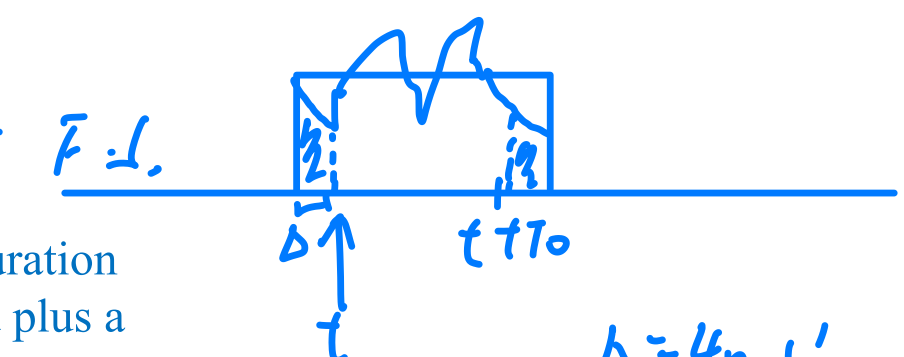
- In OFDM, the symbol time T can be quite large, e.g., ~1 ms, meaning we send only ~1000 symbols per second.
- For each symbol to convey tens of thousands of bits, we make each symbol $x_i(t)$ a very complicated signal. We make it a Fourier series sum of $N = 256$ to 2048 complex sinusoids with period $T_0 = T - 2\Delta$. The bits are encoded in the FS coefficients $\sigma_{i,k}$.
- We transmit $x_i(t)$ for a duration of $T_0 + \Delta = T - \Delta$, meaning we transmit one full period of $x_i(t)$ prepended with a cyclic prefix of duration Δ.

window from $t-\Delta$ to $t+T_0$

F.S. sum

$$x_i(t) = \sum_{k=0}^{N-1} \sigma_{i,k} e^{jk\frac{2\pi}{T_0}t}\left[u(t+\Delta) - u(t-T_0)\right]$$

IFFT N 4 F.S.



$\Delta = 4\text{ms}'$

The information waveform is a sum of complex sinusoids – a Fourier series with N terms. N can be as large as 1024 or 2048

Orthogonal sub-carriers that are T_0-periodic harmonics

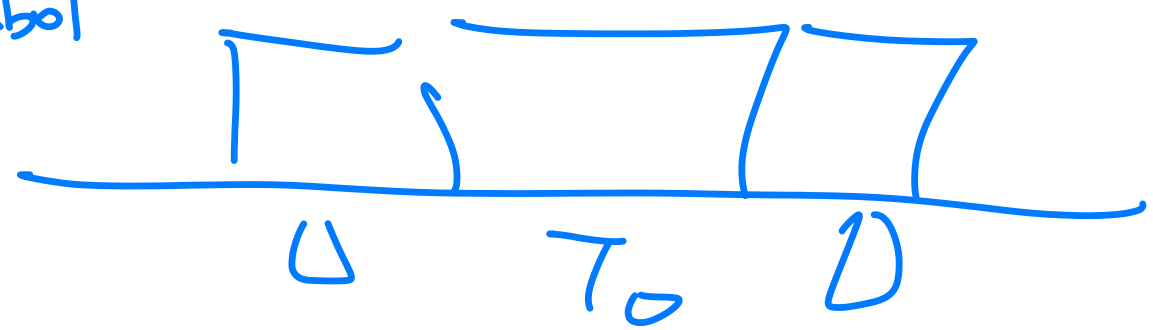
Transmit the signal for duration of $T_0 + \Delta$: one full period plus a prepended prefix

N subcarriers!

- Then a radio/optical carrier is used to transmit a sequence of symbols: $\sum_i x_i(t - iT)\, e^{j\omega_c t}$

Carrier

one symbol

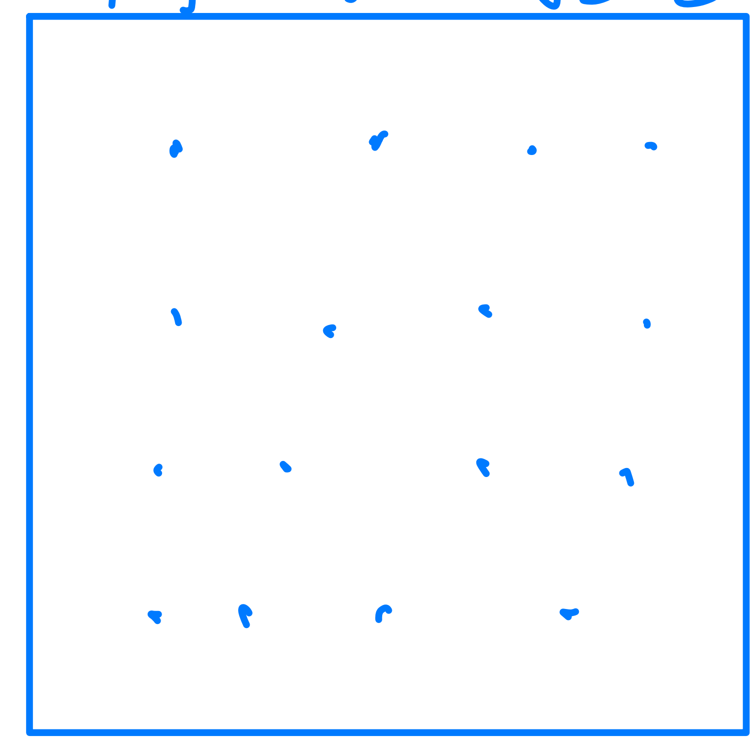


1 OFDM signal

Subcarriers . $\log_2$(QAM) bits

同一时间 send X 个 symbols ⇒ X sub-carriers.

2^4.



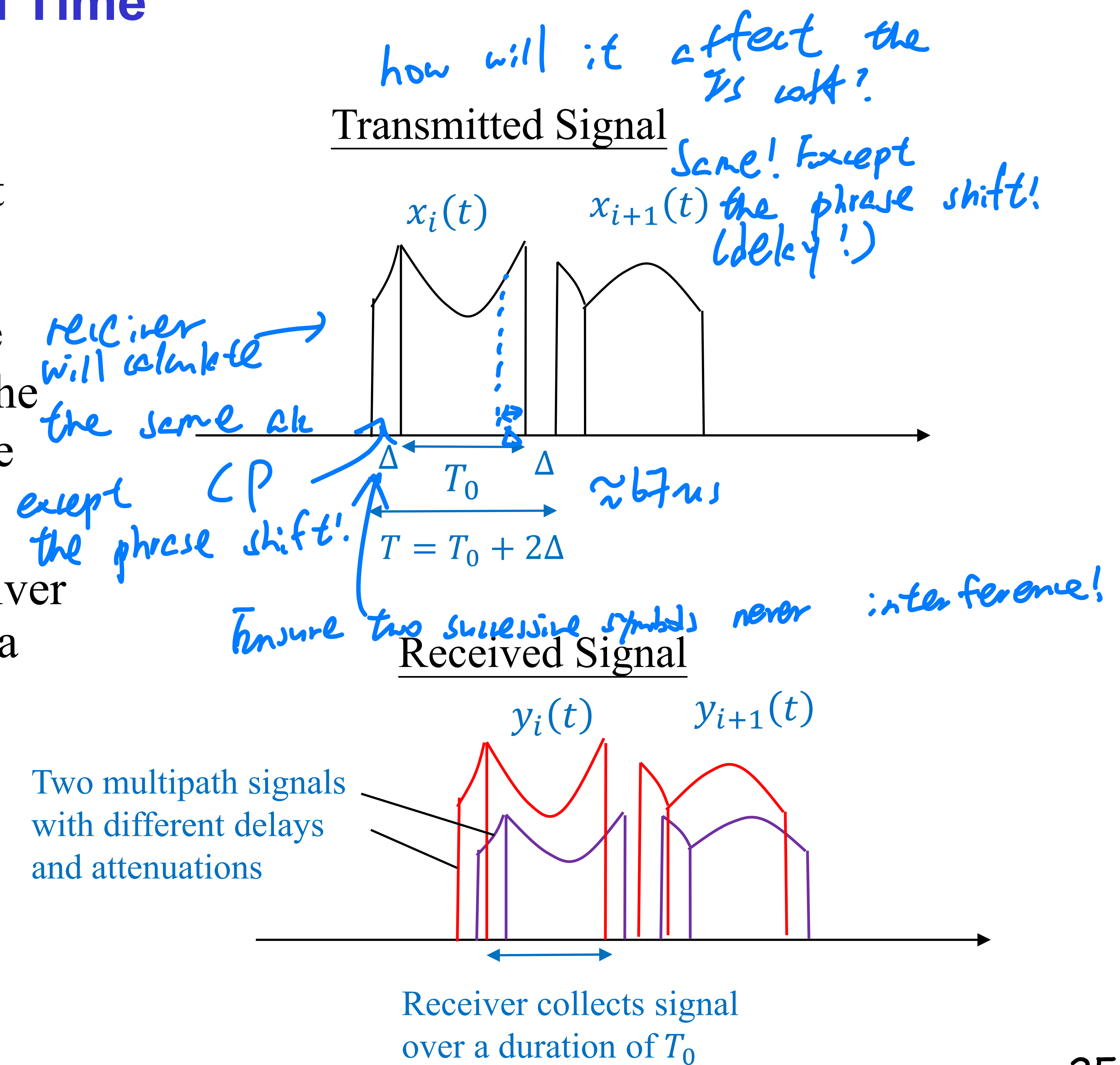
bit rate ↑

x 致 symbol rate!

T 不会改

Cyclic Prefix and Guard Time

- Receiver may receive multiple multipath signals with different delays and different attenuation
- For each symbol, the receiver collects the signal over a duration of T_0. Because of the cyclic prefix, the receiver will receive one full period of every multipath signal
- The additional guard time Δ ensures receiver will hear signal from only one symbol at a time



FS Coefficient Estimation

- The received signal is a sum of multiple copies of the transmitted signal with different delays and attenuations:

$$y_i(t) = \rho_0 x_i(t) + \rho_1 x_i(t-\tau_1) + \ldots + \rho_M x_i(t-\tau_M)$$

- Now the receiver tries to decode the bits by calculating the FS coefficients of $y_i(t)$. Let the FS coefficients of $y_i(t)$ be $\gamma_{i,k}$, then:

for kth harmonics! / subcarriers!

not change quickly with i!

$$\gamma_{i,k} = \left(\rho_0 + \rho_1 e^{-jk\frac{2\pi}{T_0}\tau_1} + \ldots + \rho_M e^{-jk\frac{2\pi}{T_0}\tau_M}\right) \sigma_{i,k} = c_k \sigma_{i,k}$$

i th symbol! attenuation phase shift

- The k-th FS coefficient is only scaled by a complex constant c_k, a channel coefficient. We can decode the bits by estimating all c_k's (by sending a pilot signal), or we can simply encode the bits by how the signal moves around the constellation from symbol to symbol for each k.
- Values of the channel coefficient c_k is known as the channel state information (CSI). Depending on whether the user is moving we may have to estimate CSI every few hundred milliseconds.

pilot!

complex number!

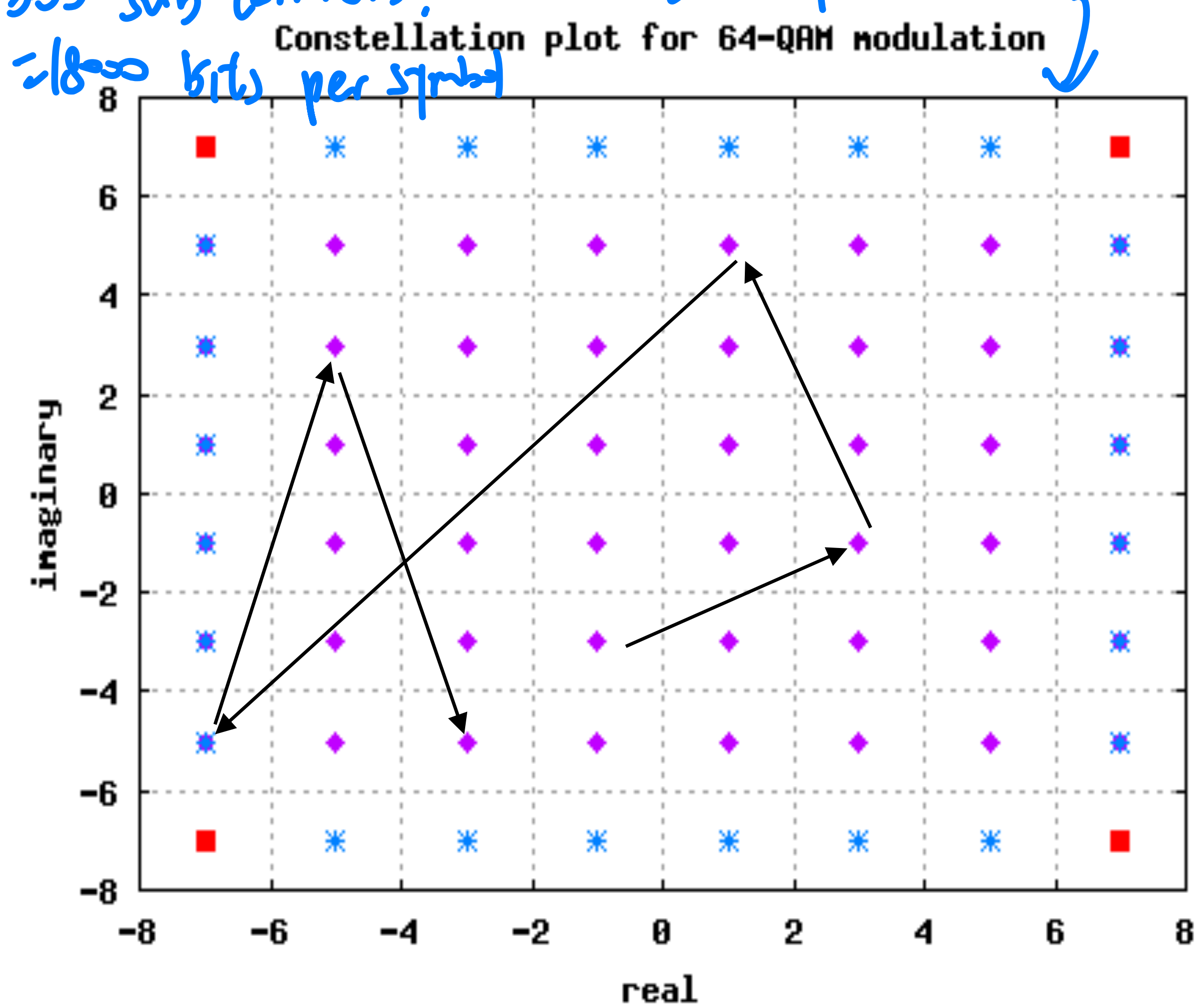
Differential Encoding

- The channel coefficient represents a scaling and rotation of the signal constellation.
- In differential encoding, the information bits are encoded by how we move around the constellation from symbol to symbol. We can decode without knowing how the channel has been scaled and rotated.

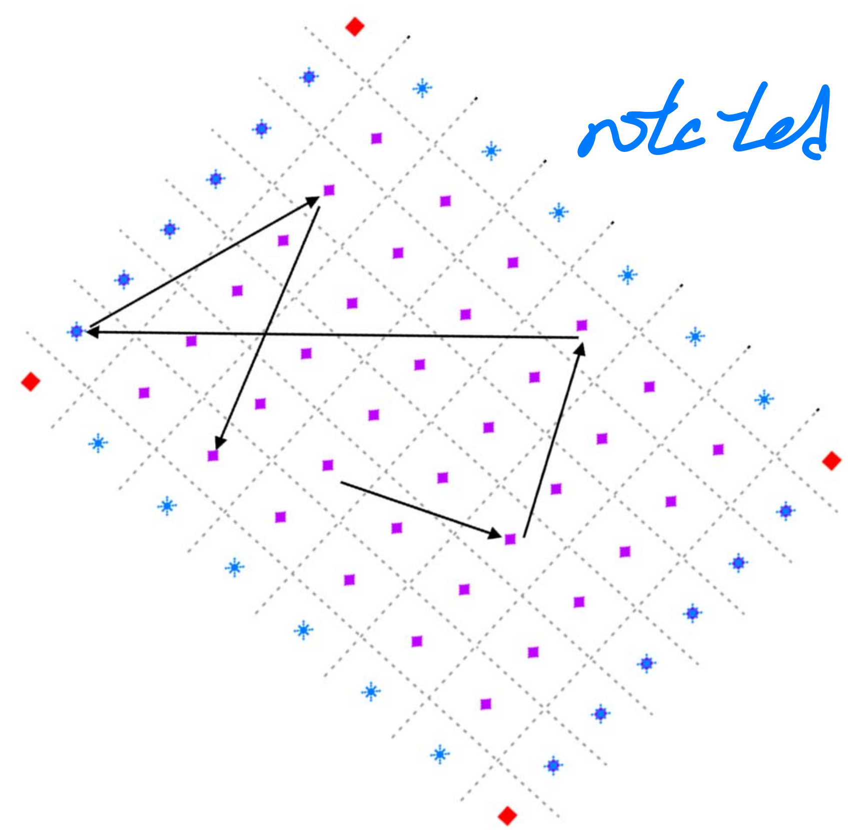
3000 sub carriers!
=18000 bits per symbol

convey 6 bits!!!

amplitude change & phase shift



rotated and scaled



might check how its amplitude change / phrase change

just opposite direction

不用 pilot!!!

encode my bit depends on how I move!