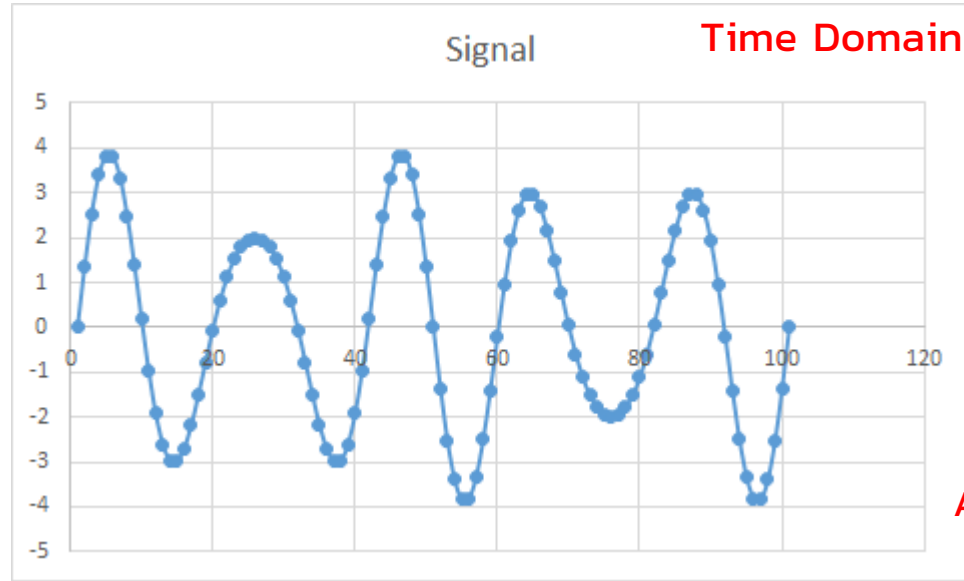


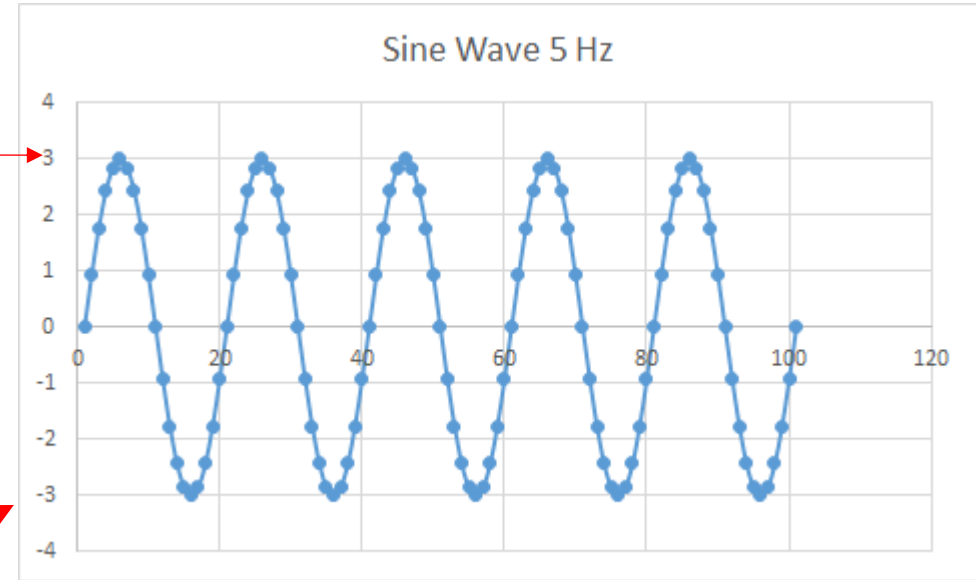
Chapter 10

4G (OFDM)

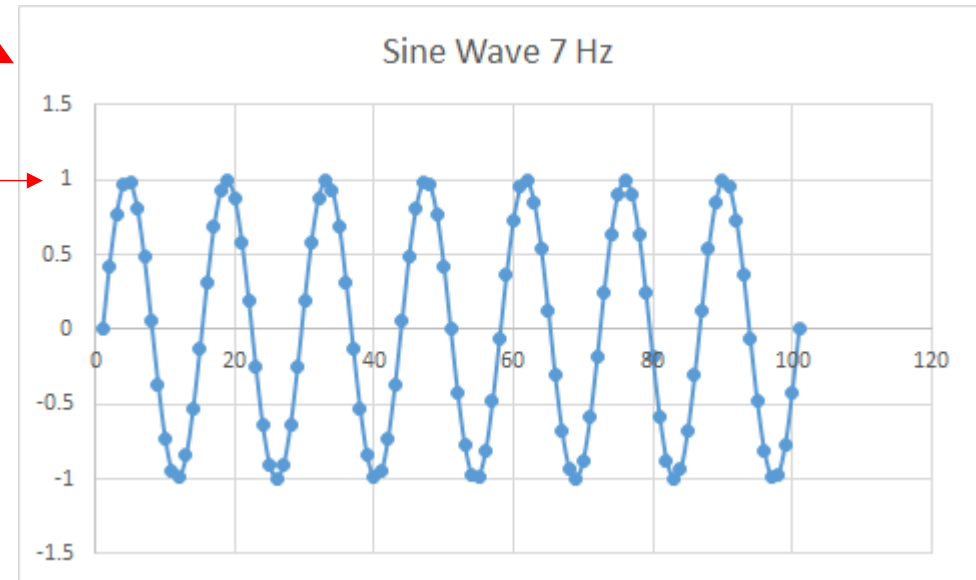
Time Domain vs. Frequency Domain



Amp. = 3

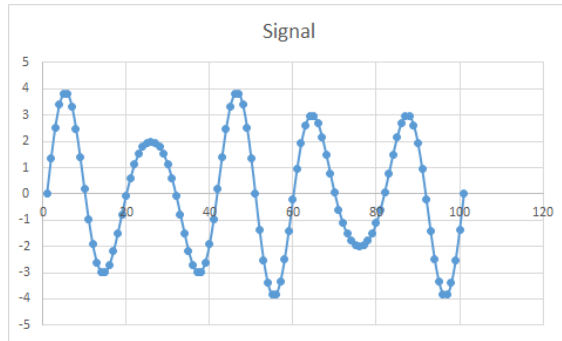


Amp. = 1

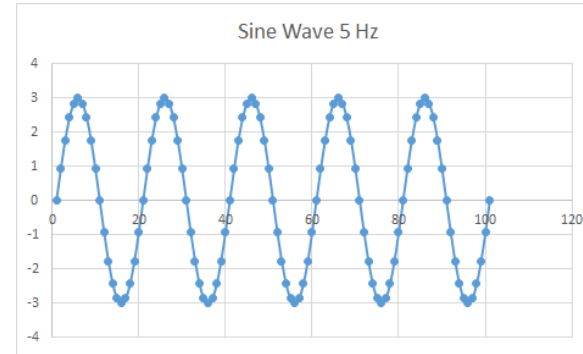


Time Domain vs. Frequency Domain

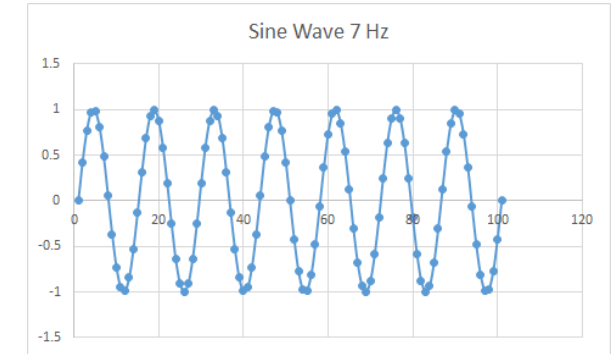
Time Domain



=



+



0.000000
0.927051
1.763356
2.427051
2.853170
3.000000
2.853170
2.427051
...

Fourier Transform



Frequency Domain

Frequency	0	1	2	3	4	5	6	7	8	9	10
Amplitude	-	-	-	-	-	3	-	1	-	-	-

Sampling rate
มีหน่วยเป็น Hz (ก็ครั้งต่อวินาที)

Discrete Fourier Transform

CHAPTER

8

The Discrete Fourier Transform

Fourier analysis is a family of mathematical techniques, all based on decomposing signals into sinusoids. The discrete Fourier transform (DFT) is the family member used with *digitized* signals. This is the first of four chapters on the real DFT, a version of the discrete Fourier transform that uses real numbers to represent the input and output signals. The **complex DFT**, a more advanced technique that uses complex numbers, will be discussed in Chapter 31. In this chapter we look at the mathematics and algorithms of the Fourier decomposition, the heart of the DFT.

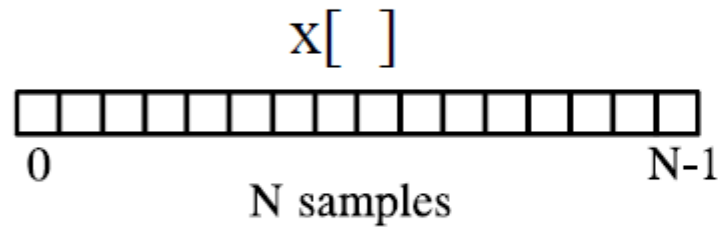
The Family of Fourier Transform

Fourier analysis is named after **Jean Baptiste Joseph Fourier** (1768-1830), a French mathematician and physicist. (Fourier is pronounced: for-ē-ā, and is always capitalized). While many contributed to the field, Fourier is honored for his mathematical discoveries and insight into the practical usefulness of the techniques. Fourier was interested in heat propagation, and presented a paper in 1807 to the Institut de France on the use of sinusoids to represent temperature distributions. The paper contained the controversial claim that any continuous periodic signal could be represented as the sum of properly chosen sinusoidal waves. Among the reviewers were two of history's most famous mathematicians, Joseph Louis Lagrange (1736-1813), and Pierre Simon de Laplace (1749-1827).

While Laplace and the other reviewers voted to publish the paper, Lagrange adamantly protested. For nearly 50 years, Lagrange had insisted that such an approach could not be used to represent signals with *corners*, i.e., discontinuous slopes, such as in square waves. The Institut de France bowed to the prestige of Lagrange, and rejected Fourier's work. It was only after Lagrange died that the paper was finally published, some 15 years later. Luckily, Fourier had other things to keep him busy, political activities, expeditions to Egypt with Napoleon, and trying to avoid the guillotine after the French Revolution (literally!).

Forward DFT vs. Inverse DFT

Time Domain

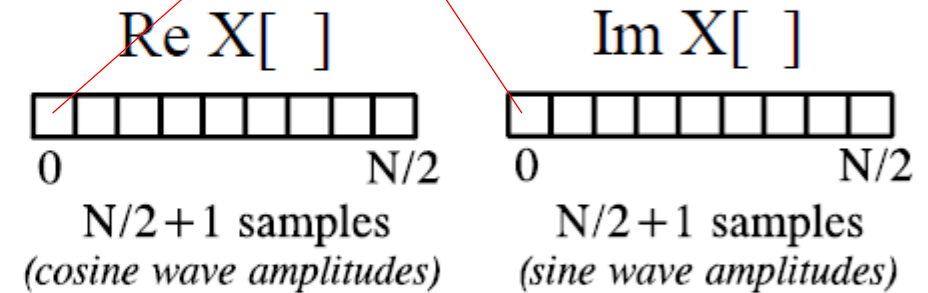


Forward DFT

Inverse DFT

Frequency Domain

ความถี่เท่ากัน ต่างกันที่เป็น cosine หรือ sine



collectively referred to as $X[\]$

Coding: find $\text{Re}X[]$ and $\text{Im}X$

$$\text{Re}X[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi k i / N)$$

$$\text{Im}X[k] = - \sum_{i=0}^{N-1} x[i] \sin(2\pi k i / N)$$

Coding: convert $\text{Re}X[]$ & $\text{Im}X[]$ to $\text{Re}\bar{X}[]$ & $\text{Im}\bar{X}[]$

$$\text{Re}\bar{X}[k] = \frac{\text{Re}X[k]}{N/2}$$

$$\text{Im}\bar{X}[k] = -\frac{\text{Im}X[k]}{N/2}$$

except for two special cases:

$$\text{Re}\bar{X}[0] = \frac{\text{Re}X[0]}{N}$$

$$\text{Re}\bar{X}[N/2] = \frac{\text{Re}X[N/2]}{N}$$

$$\text{Re}\bar{X}[k]$$

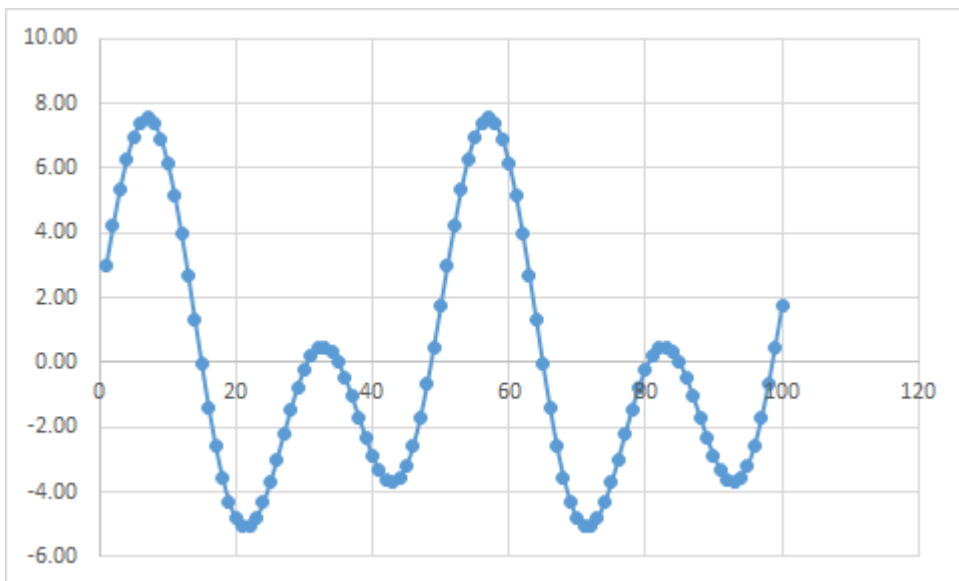
คือ amplitude ของ cosine wave ที่มีความถี่ $(k/N) \times \text{sampling rate}$

$$\text{Im}\bar{X}[k]$$

คือ amplitude ของ sine wave ที่มีความถี่ $(k/N) \times \text{sampling rate}$

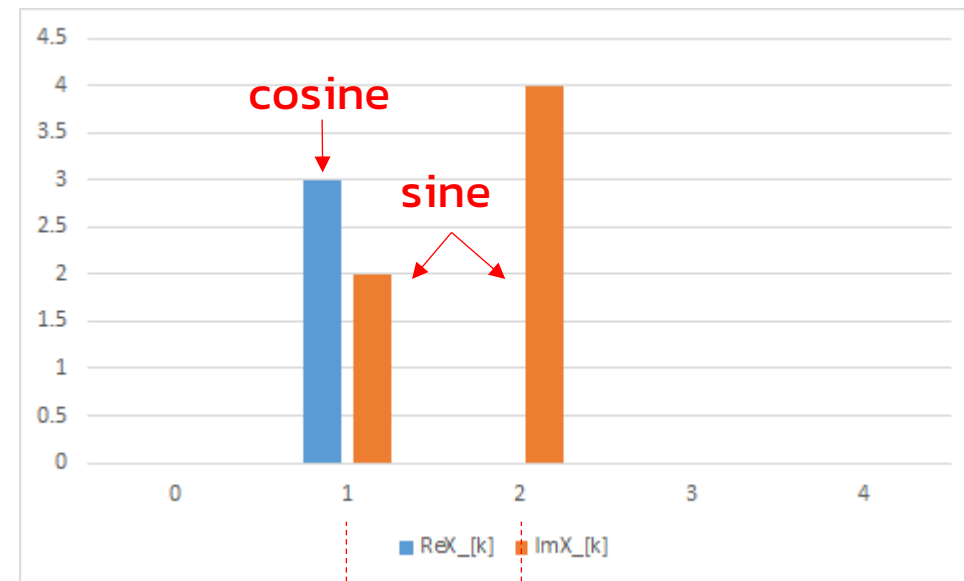
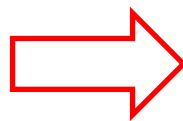
ถ้าอยากเห็นความถี่ f ต้องใช้ sample rate
อย่างน้อย 2 เท่าของ f เพราะ k มีค่าได้มากที่สุด $N/2$
ที่ช่อง $k = N/2$ คือ ความถี่ $\text{sampling rate} / 2$

Example



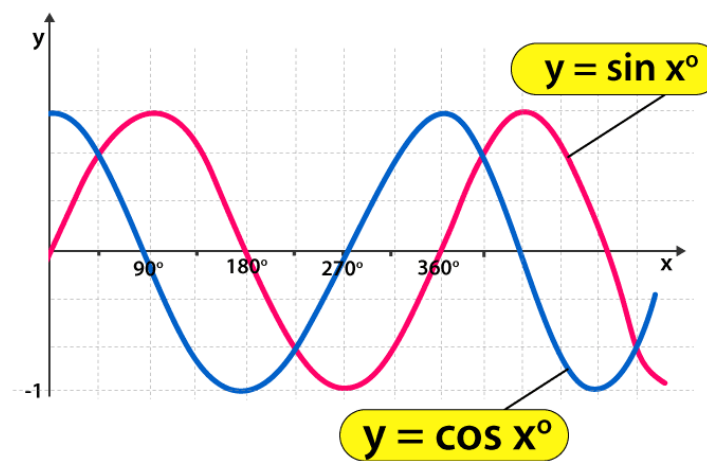
$$y = 3 \cdot \cos(2\pi \cdot f_1 \cdot t) + 2 \cdot \sin(2\pi \cdot f_1 \cdot t) + 4 \cdot \sin(2\pi \cdot f_2 \cdot t)$$

Forward DFT



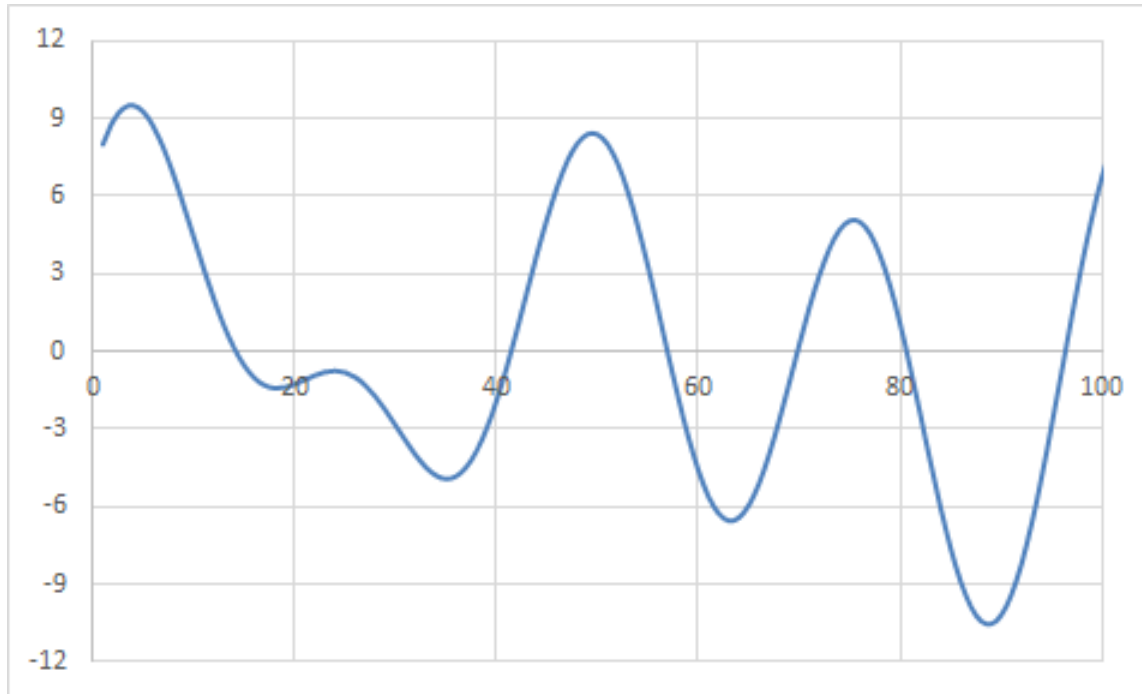
$k = 1$ $k = 2$

$f = (k/N) * \text{sampling rate}$



Assignment:

จงแยกความถี่ของสัญญาณต่อไปนี้ ว่าประกอบด้วยความถี่อะไรบ้าง เป็น cosine และ sine ที่มี amplitude เท่าใด

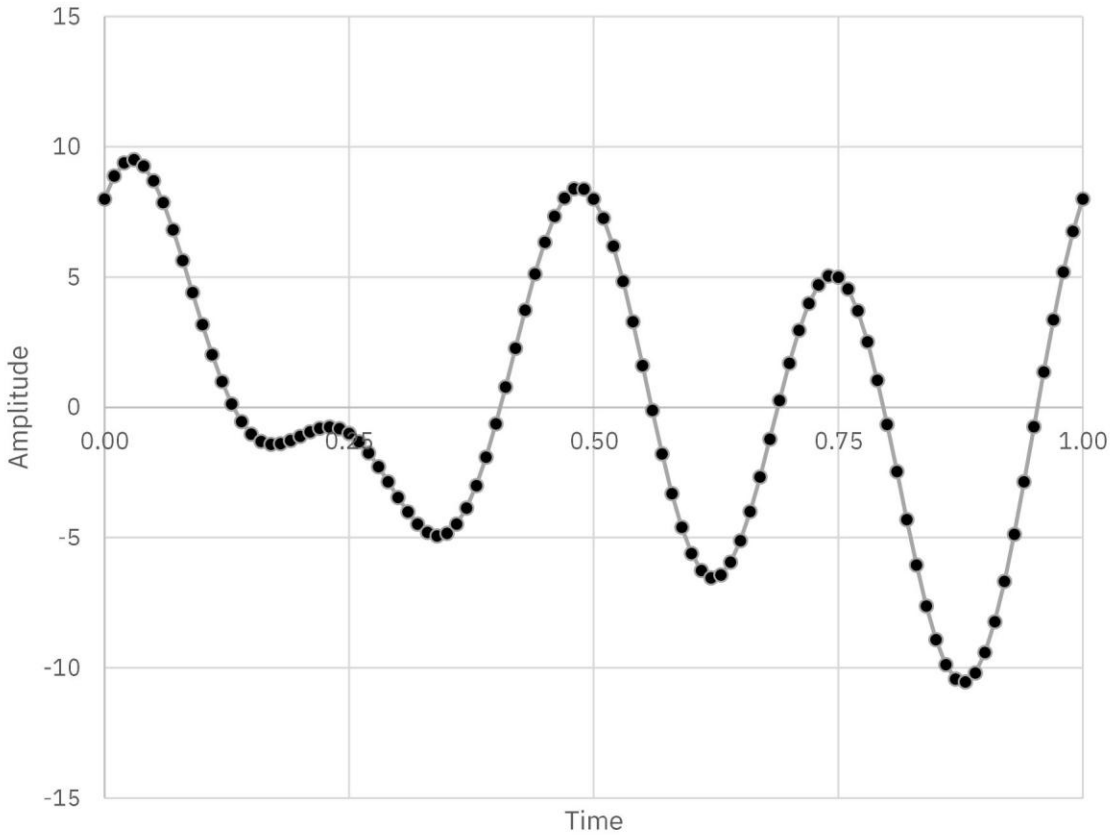


0.68	1.218220
0.69	0.269492
0.70	1.693689
0.71	2.361013
0.72	3.986797
0.73	4.700155
0.74	5.048360
0.75	5.000000
0.76	4.547028
0.77	3.709399
0.78	2.514289
0.79	1.033998
0.80	-0.657452
0.81	-2.468707
0.82	-4.300273
0.83	-6.050177
0.84	-7.619773
0.85	-8.919334
0.86	-9.871129
0.87	-10.423663
0.88	-10.534855
0.89	-10.193064
0.90	-9.412338
0.91	-8.223722
0.92	-6.688933
0.93	-4.865773
0.94	-2.855668
0.95	-0.748520
0.96	1.357668
0.97	3.367235
0.98	5.192072
0.99	6.746278
1.00	8.000000

ข้อมูลอยู่ในไฟล์ Excel (sampling rate = 100 Hz)

ดูเฉลยในหน้าถัดไป

t	$x(t)$	t	$x(t)$	t	$x(t)$	t	$x(t)$
0.00	8.0000	0.25	-1.0000	0.50	8.0000	0.75	5.0000
0.01	8.8822	0.26	-1.3152	0.51	7.2576	0.76	4.5470
0.02	9.3825	0.27	-1.7486	0.52	6.1868	0.77	3.7054
0.03	9.5011	0.28	-2.2758	0.53	4.8397	0.78	2.5143
0.04	9.2584	0.29	-2.8606	0.54	3.2847	0.79	1.0340
0.05	8.6928	0.30	-3.4576	0.55	1.6026	0.80	-0.6575
0.06	7.8574	0.31	-4.0154	0.56	-0.1175	0.81	-2.4687
0.07	6.8165	0.32	-4.4801	0.57	-1.7837	0.82	-4.3003
0.08	5.6411	0.33	-4.7999	0.58	-3.3066	0.83	-6.0502
0.09	4.4042	0.34	-4.9285	0.59	-4.6044	0.84	-7.6198
0.10	3.1761	0.35	-4.8292	0.60	-5.6079	0.85	-8.9193
0.11	2.0205	0.36	-4.4781	0.61	-6.2648	0.86	-9.8731
0.12	0.9905	0.37	-3.8663	0.62	-6.5427	0.87	-10.4237
0.13	0.1258	0.38	-3.0017	0.63	-6.4316	0.88	-10.5349
0.14	-0.5489	0.39	-1.9087	0.64	-5.9440	0.89	-10.1940
0.15	-1.0249	0.40	-0.6281	0.65	-5.1151	0.90	-9.4122
0.16	-1.3091	0.41	0.7848	0.66	-4.0005	0.91	-8.2237
0.17	-1.4226	0.42	2.2638	0.67	-2.6729	0.92	-6.6839
0.18	-1.3981	0.43	3.7345	0.68	-1.2182	0.93	-4.8658
0.19	-1.2772	0.44	5.1192	0.69	0.2695	0.94	-2.8557
0.20	-1.1065	0.45	6.3417	0.70	1.6937	0.95	-0.7485
0.21	-0.9336	0.46	7.3314	0.71	2.9610	0.96	1.3577
0.22	-0.8033	0.47	8.0286	0.72	3.9868	0.97	3.3672
0.23	-0.7538	0.48	8.3877	0.73	4.7002	0.98	5.1921
0.24	-0.8139	0.49	8.3809	0.74	5.0484	0.99	6.7563
						1.00	8.0000



t	$ReX(t)$	$ImX(t)$	t	$ReX(t)$	$ImX(t)$
0.00	0	0	0.26	0	0
0.01	0	-50	0.27	0	0
0.02	150	-100	0.28	0	0
0.03	0	-200	0.29	0	0
0.04	250	0	0.30	0	0
0.05	0	0	0.31	0	0
0.06	0	0	0.32	0	0
0.07	0	0	0.33	0	0
0.08	0	0	0.34	0	0
0.09	0	0	0.35	0	0
0.10	0	0	0.36	0	0
0.11	0	0	0.37	0	0
0.12	0	0	0.38	0	0
0.13	0	0	0.39	0	0
0.14	0	0	0.40	0	0
0.15	0	0	0.41	0	0
0.16	0	0	0.42	0	0
0.17	0	0	0.43	0	0
0.18	0	0	0.44	0	0
0.19	0	0	0.45	0	0
0.20	0	0	0.46	0	0
0.21	0	0	0.47	0	0
0.22	0	0	0.48	0	0
0.23	0	0	0.49	0	0
0.24	0	0	0.50	0	0
0.25	0	0			

$$ReX[k] = \sum_{i=0}^{N-1} x[i] \cos\left(\frac{2\pi ki}{N}\right), \quad 0 \leq k \leq \frac{N}{2}$$

$$ImX[k] = - \sum_{i=0}^{N-1} x[i] \sin\left(\frac{2\pi ki}{N}\right), \quad 0 \leq k \leq \frac{N}{2}$$

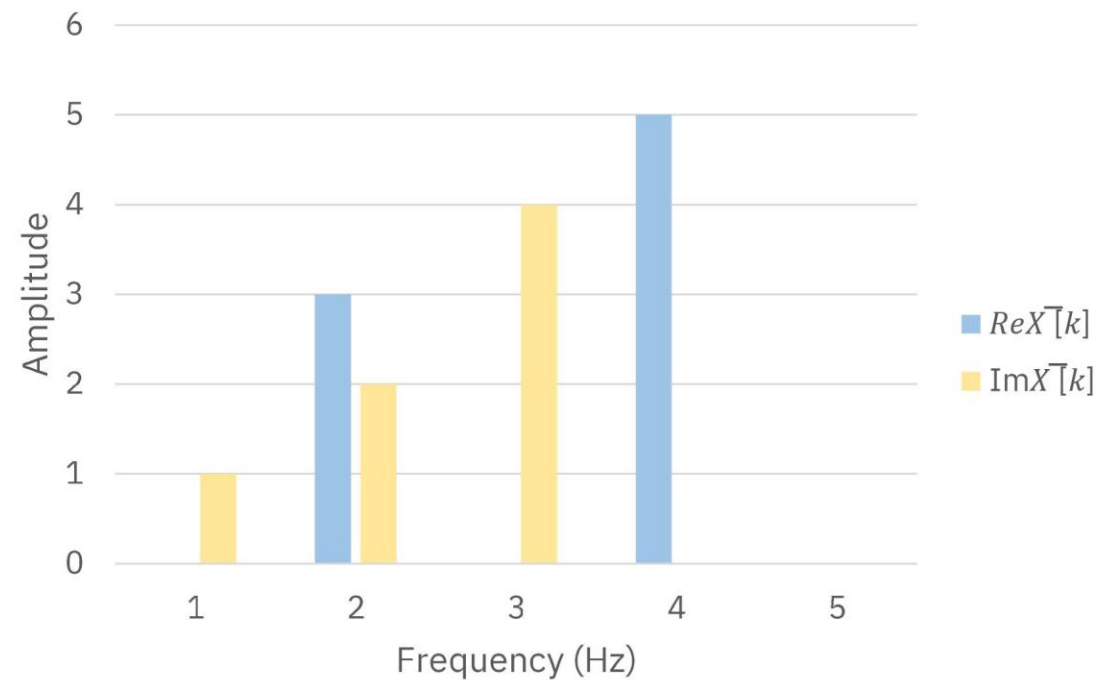
t	$Re\bar{X}[k]$	$Im\bar{X}[k]$	$frequency$	t	$Re\bar{X}[k]$	$Im\bar{X}[k]$	$frequency$
0.00	0	0	0	0.26	0	0	26
0.01	0	1	1	0.27	0	0	27
0.02	3	2	2	0.28	0	0	28
0.03	0	4	3	0.29	0	0	29
0.04	5	0	4	0.30	0	0	30
0.05	0	0	5	0.31	0	0	31
0.06	0	0	6	0.32	0	0	32
0.07	0	0	7	0.33	0	0	33
0.08	0	0	8	0.34	0	0	34
0.09	0	0	9	0.35	0	0	35
0.10	0	0	10	0.36	0	0	36
0.11	0	0	11	0.37	0	0	37
0.12	0	0	12	0.38	0	0	38
0.13	0	0	13	0.39	0	0	39
0.14	0	0	14	0.40	0	0	40
0.15	0	0	15	0.41	0	0	41
0.16	0	0	16	0.42	0	0	42
0.17	0	0	17	0.43	0	0	43
0.18	0	0	18	0.44	0	0	44
0.19	0	0	19	0.45	0	0	45
0.20	0	0	20	0.46	0	0	46
0.21	0	0	21	0.47	0	0	47
0.22	0	0	22	0.48	0	0	48
0.23	0	0	23	0.49	0	0	49
0.24	0	0	24	0.50	0	0	50
0.25	0	0	25				

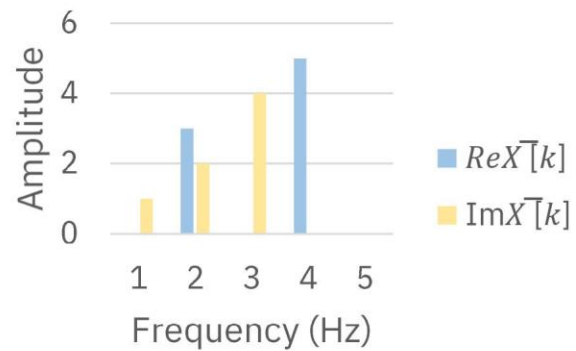
$$Re\bar{X}[k] = \begin{cases} \frac{ReX[0]}{N}, & k = 0 \\ \frac{ReX[\frac{N}{2}]}{N}, & k = \frac{N}{2} \\ \frac{ReX[k]}{\frac{N}{2}}, & otherwise \end{cases}$$

$$Im\bar{X}[k] = \frac{-ImX[k]}{\frac{N}{2}}$$

$$Frequency[k] = \frac{k}{N} \times Sample\ Rate$$

t	$Re\bar{X}[k]$	$Im\bar{X}[k]$	$frequency$	t	$Re\bar{X}[k]$	$Im\bar{X}[k]$	$frequency$
0.00	0	0	0	0.26	0	0	26
0.01	0	1	1	0.27	0	0	27
0.02	3	2	2	0.28	0	0	28
0.03	0	4	3	0.29	0	0	29
0.04	5	0	4	0.30	0	0	30
0.05	0	0	5	0.31	0	0	31
0.06	0	0	6	0.32	0	0	32
0.07	0	0	7	0.33	0	0	33
0.08	0	0	8	0.34	0	0	34
0.09	0	0	9	0.35	0	0	35
0.10	0	0	10	0.36	0	0	36
0.11	0	0	11	0.37	0	0	37
0.12	0	0	12	0.38	0	0	38
0.13	0	0	13	0.39	0	0	39
0.14	0	0	14	0.40	0	0	40
0.15	0	0	15	0.41	0	0	41
0.16	0	0	16	0.42	0	0	42
0.17	0	0	17	0.43	0	0	43
0.18	0	0	18	0.44	0	0	44
0.19	0	0	19	0.45	0	0	45
0.20	0	0	20	0.46	0	0	46
0.21	0	0	21	0.47	0	0	47
0.22	0	0	22	0.48	0	0	48
0.23	0	0	23	0.49	0	0	49
0.24	0	0	24	0.50	0	0	50
0.25	0	0	25				





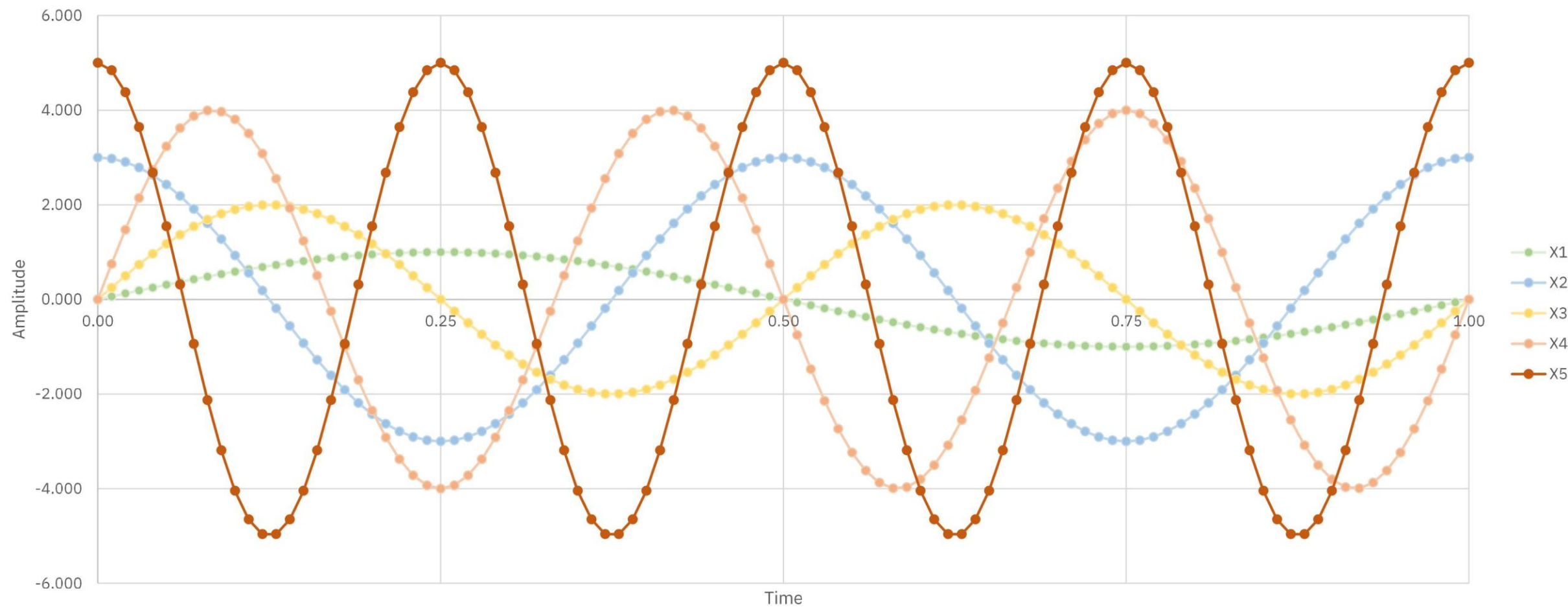
$$X_1(t) = 1 \sin(2\pi t)$$

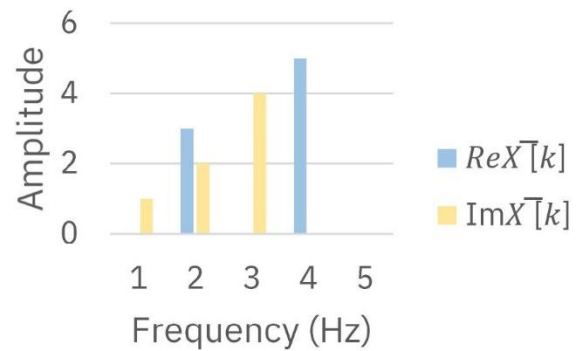
$$X_4(t) = 4 \sin(6\pi t) \quad \text{\color{red}\#5}$$

$$X_2(t) = 3 \cos(4\pi t)$$

$$X_5(t) = 5 \sin(8\pi t)$$

$$X_3(t) = 2 \sin(4\pi t)$$





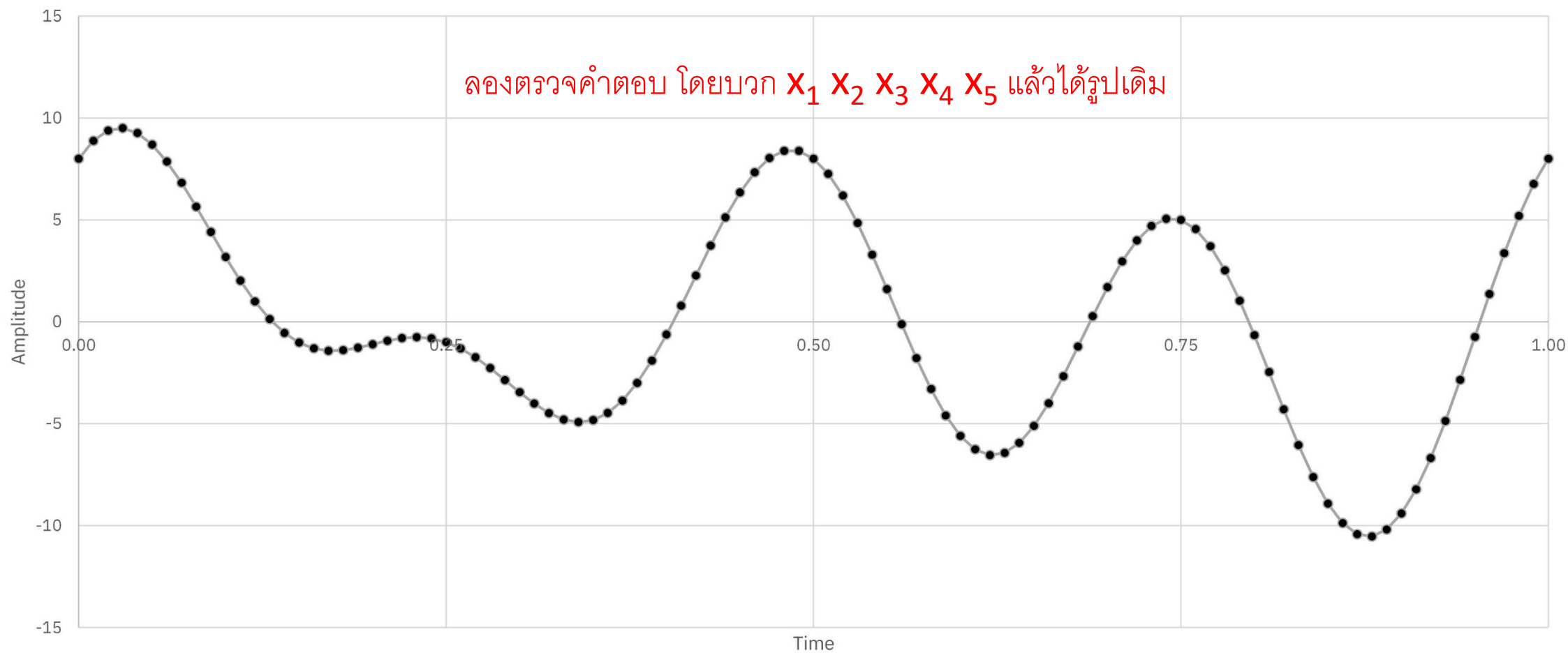
$$X_1(t) = 1 \sin(2\pi t)$$

$$X_4(t) = 4 \sin(6\pi t) \quad \text{\color{red}\#6}$$

$$X_2(t) = 3 \cos(4\pi t)$$

$$X_5(t) = 5 \sin(8\pi t)$$

$$X_3(t) = 2 \sin(4\pi t)$$



Orthogonal Frequency Division Multiplexing (OFDM)

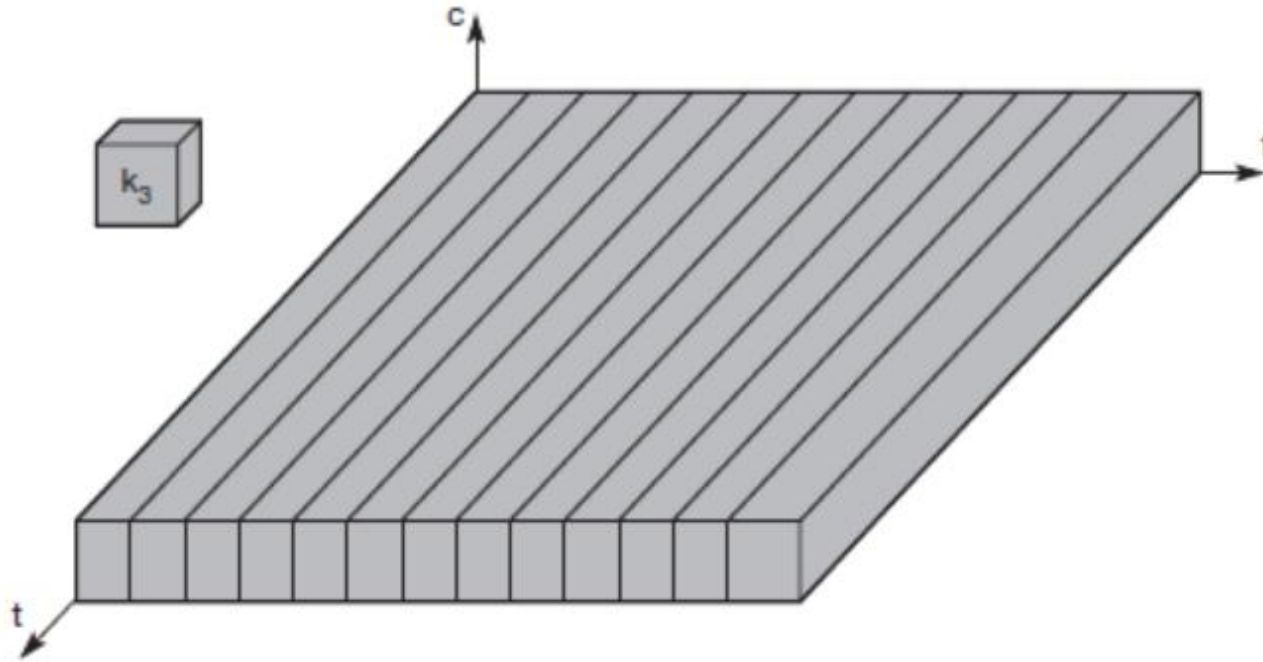


Figure 2.30
Parallel data
transmission on
several subcarriers
with lower rate

ปริมาณข้อมูลมาก ใช้ subcarriers เดียวไม่พอ ต้องใช้หลาย ๆ sub. ช่วยกันส่งข้อมูล OFDM ใช้กับ DAB, DVB, 4G แต่ละความถี่อยู่ชิดกันมาก ห่างกันแค่ 15 kHz ทำให้ได้ช่องสัญญาณจำนวนมาก ในระบบ 2G ต้องเว้นระยะมากถึง 200 kHz

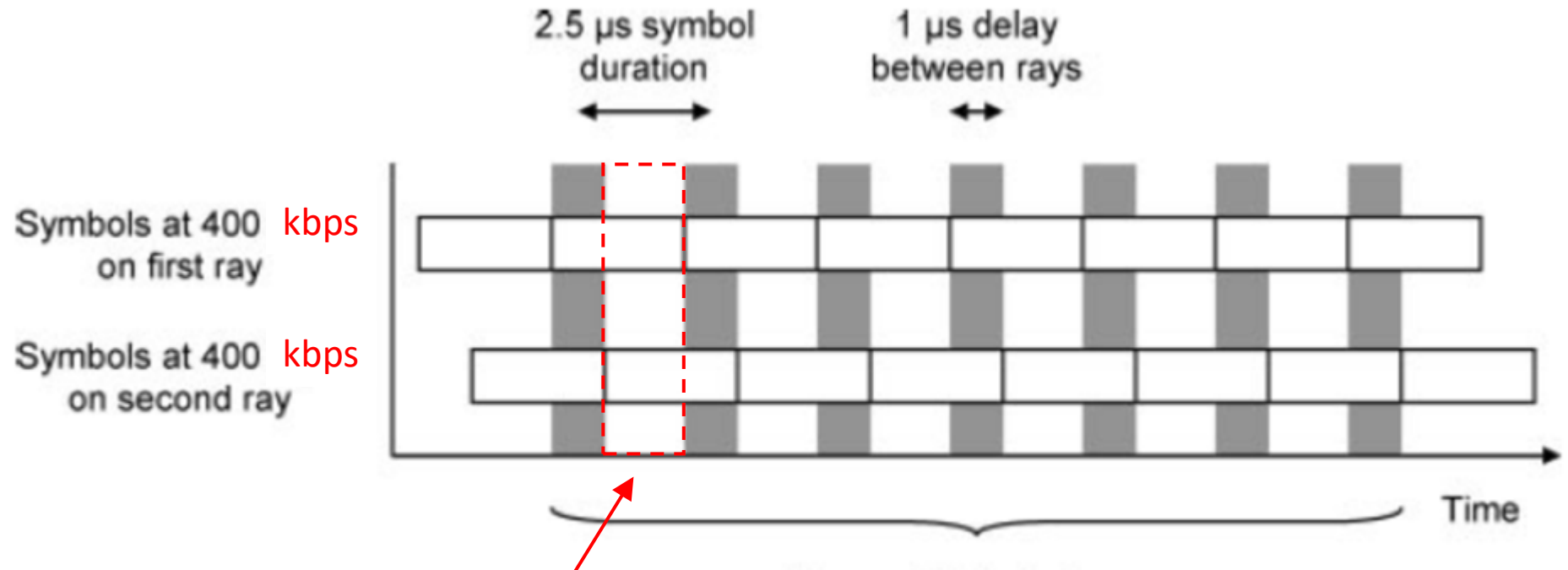
เช่น ประมวลความถี่กว้าง 15 MHz (1,710 – 1,725 MHz หรือ 1,805 – 1,820 MHz)

ระบบ 2G ใช้ FDM ได้ $15 \text{ MHz} / 200 \text{ kHz} = 75 \text{ channels}$

ระบบ 4G ใช้ OFDM ได้ $15 \text{ MHz} / 15 \text{ kHz} = 1,000 \text{ channels}$ (13 เท่า)

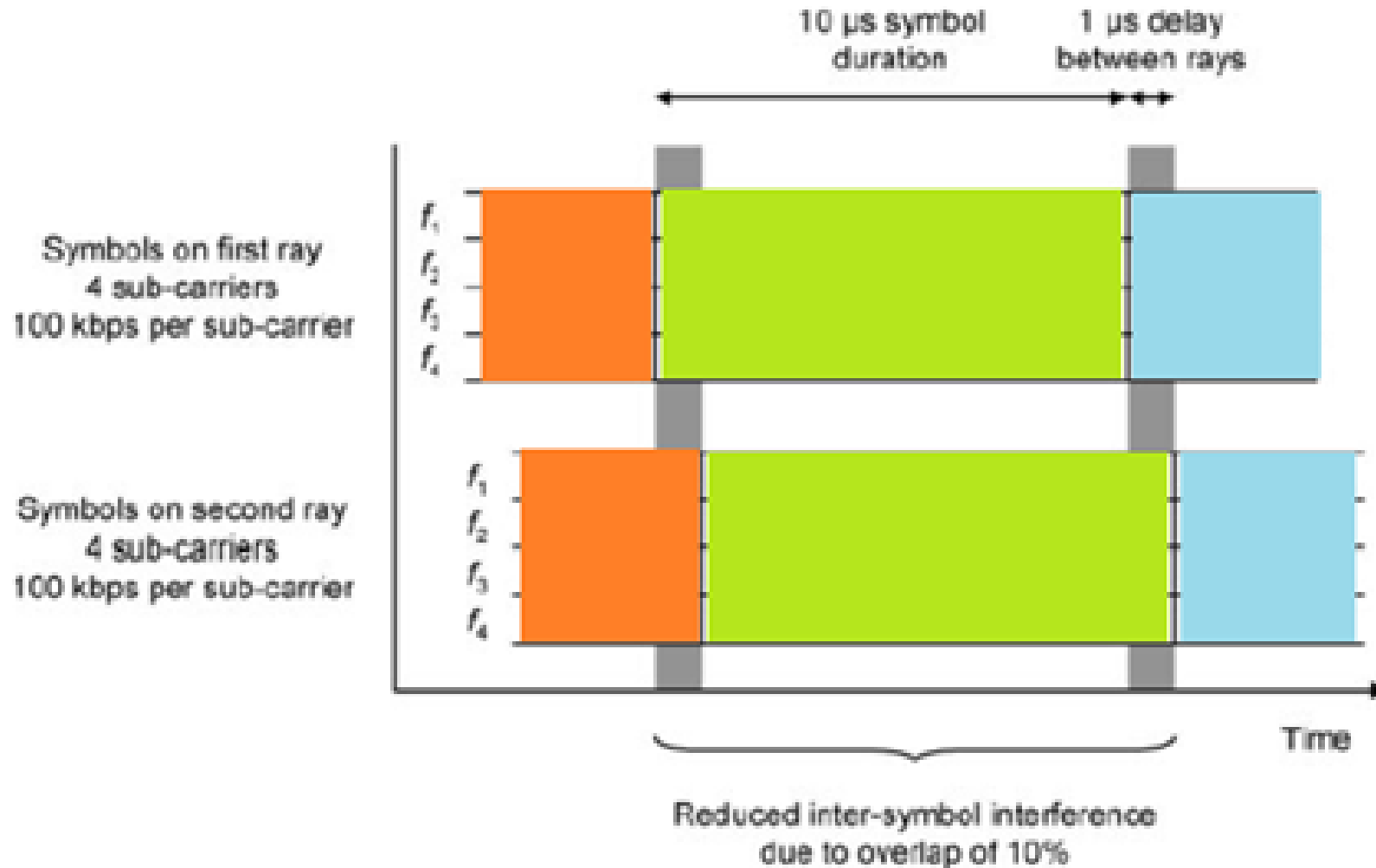
Why OFDM?

- สัญญาณคลื่นปล่อยออกมาจากแหล่งกำเนิดเดียวกัน แต่ first ray มาจาก short path ส่วน second ray มาจาก long path (มาถึงช้ากว่า)
- ส่วนที่ overlap กัน (มากถึง 40%) จะทำให้เกิด inter-symbol interference (ISI)



Symbol duration ต้องยาวพอสมควร เพื่อให้มีช่วงที่ไม่ overlap กัน
ดังนั้นจะเพิ่ม data rate โดยลด symbol duration ไม่ได้

- ใช้ 4 sub-carriers (frequencies) ถ้ารักษา data rate ให้เท่าเดิม symbol duration จะยาวขึ้น 4 เท่า จาก $2.5 \mu\text{s}$ เป็น $10 \mu\text{s}$
- ส่วนที่ overlap กันน้อยลงเหลือไม่เกิน 10% (ส่วนที่ overlap ระหว่าง symbol ที่ต่างกัน - ดูสีที่ต่างกัน)

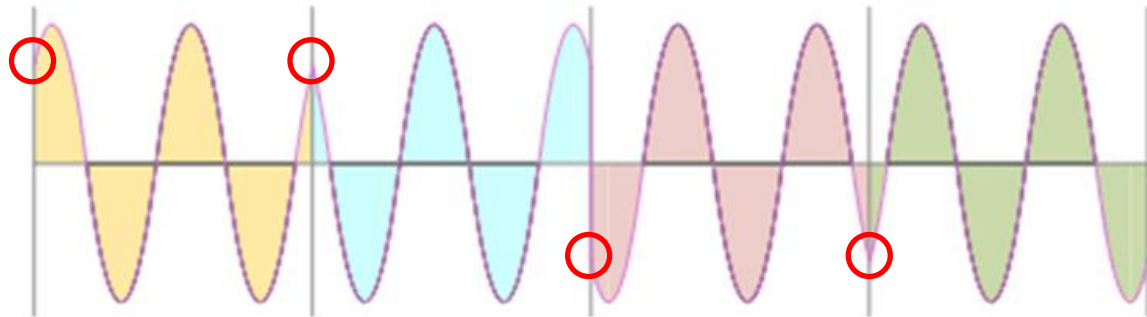


จะเพิ่ม data rate โดยลด symbol duration ไม่ได้ ก็ใช้วิธีเพิ่มจำนวน subcarrier แทน

หรือไม่ต้องการเพิ่ม data rate ก็เพิ่มจำนวน subcarrier ได้ เพื่อลด inter-symbol interference

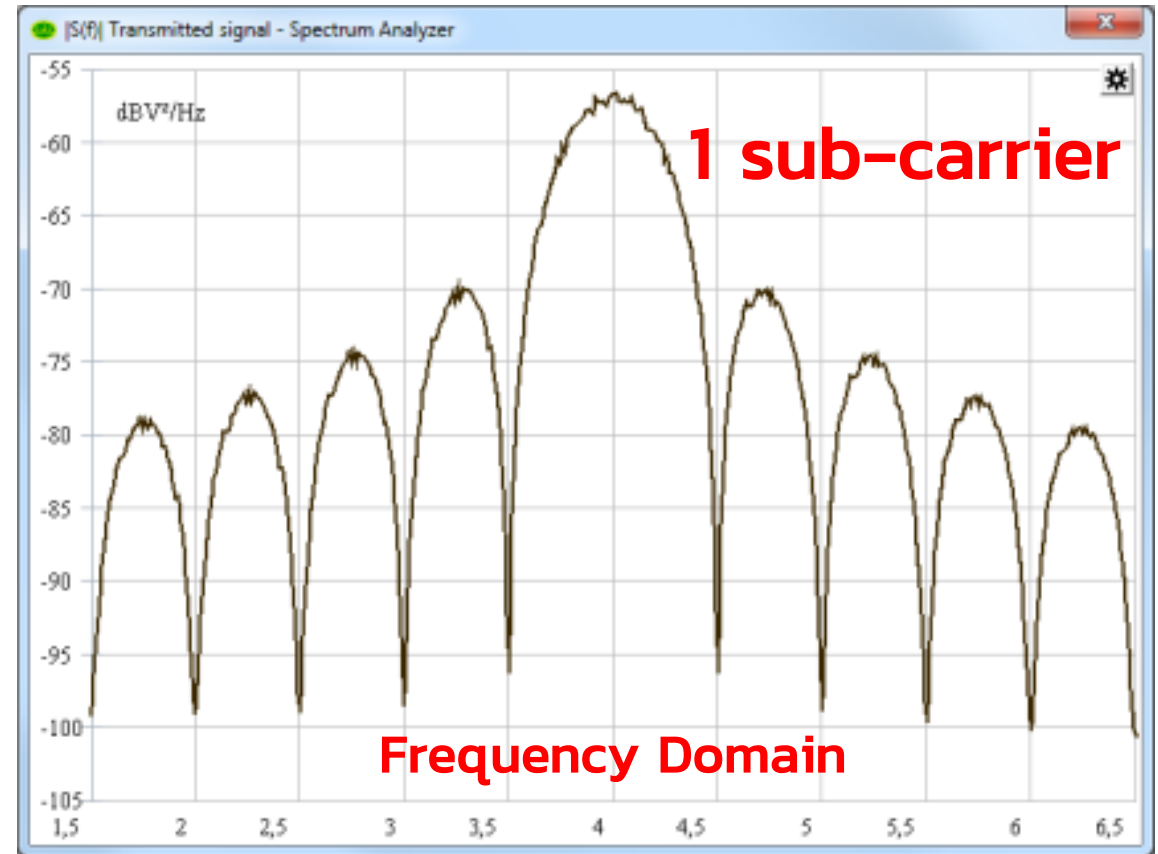
QPSK Spectrum

ข้อมูลที่เข้ารหัสด้วย QPSK

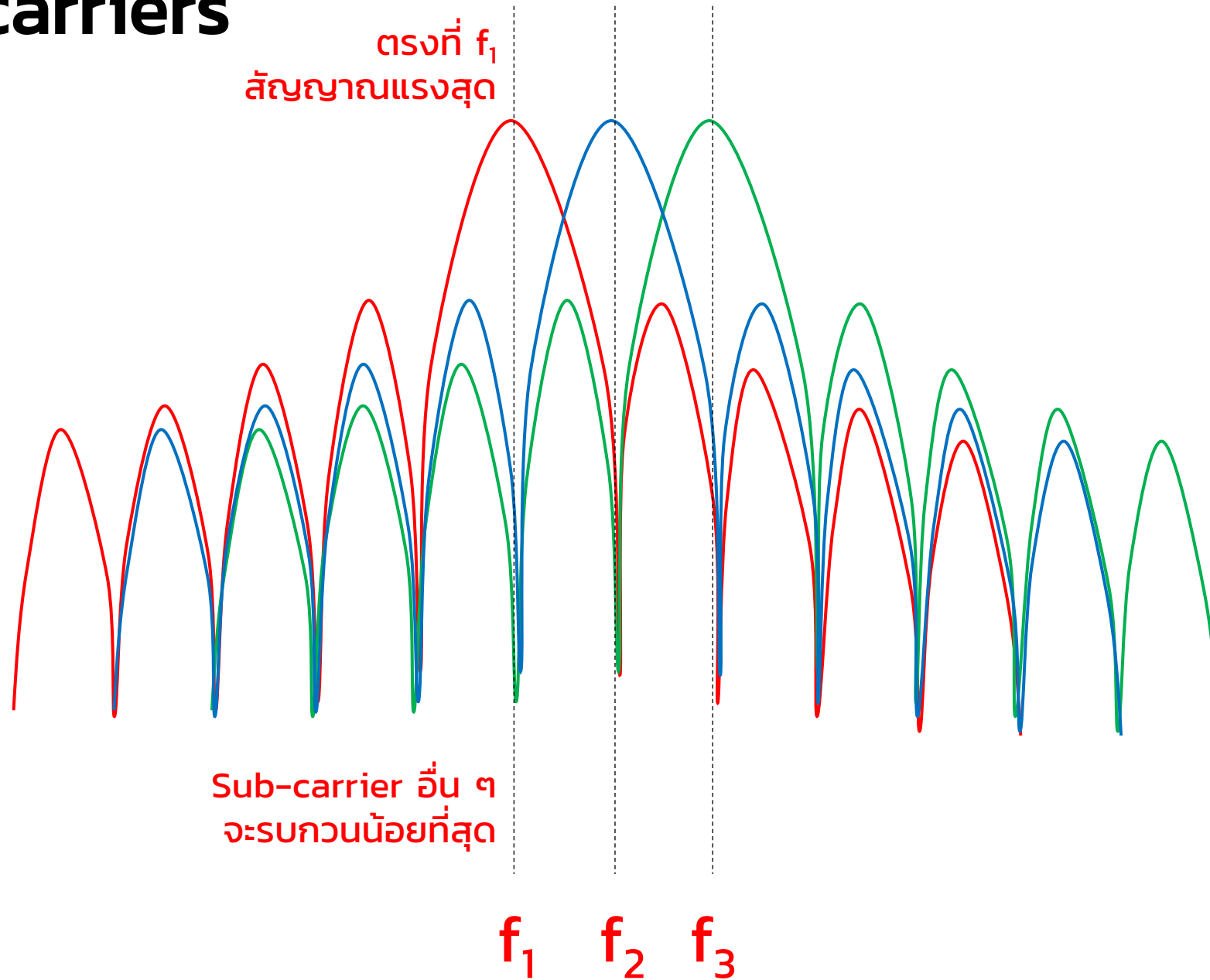


Time Domain

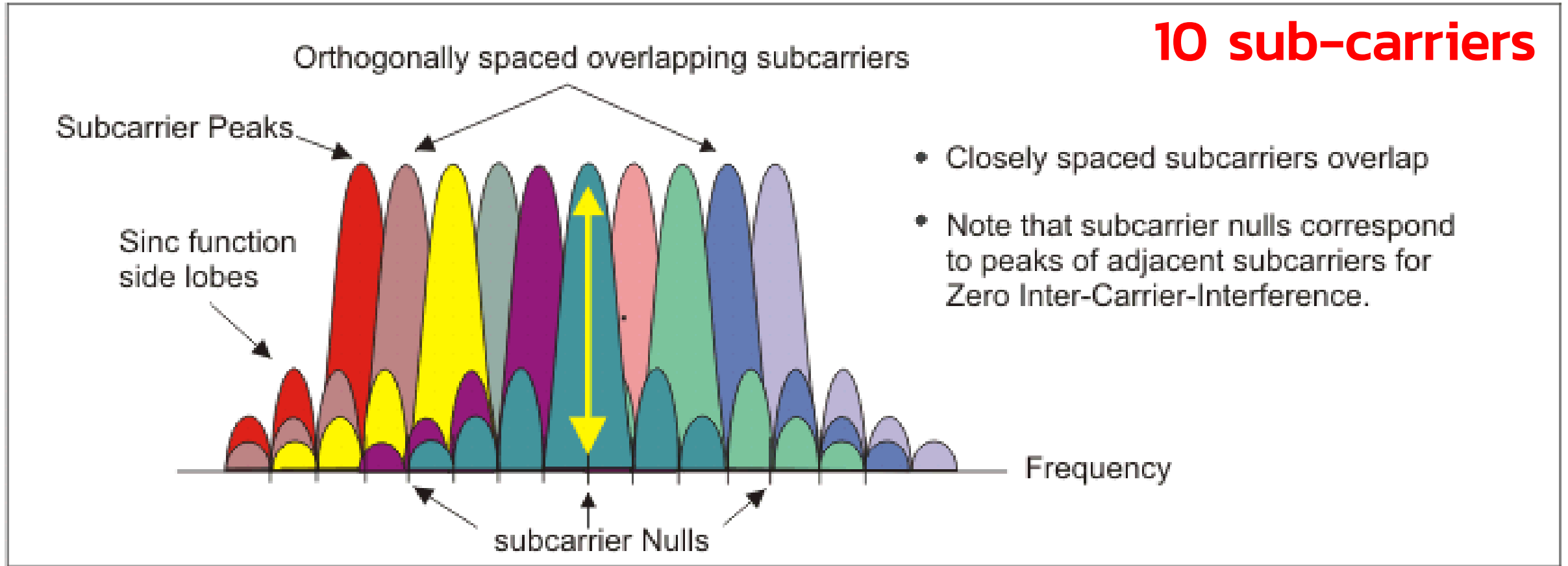
ใช้ random data และ QPSK
แล้วทำ Forward DFT จะได้รูปนี้



3 sub-carriers

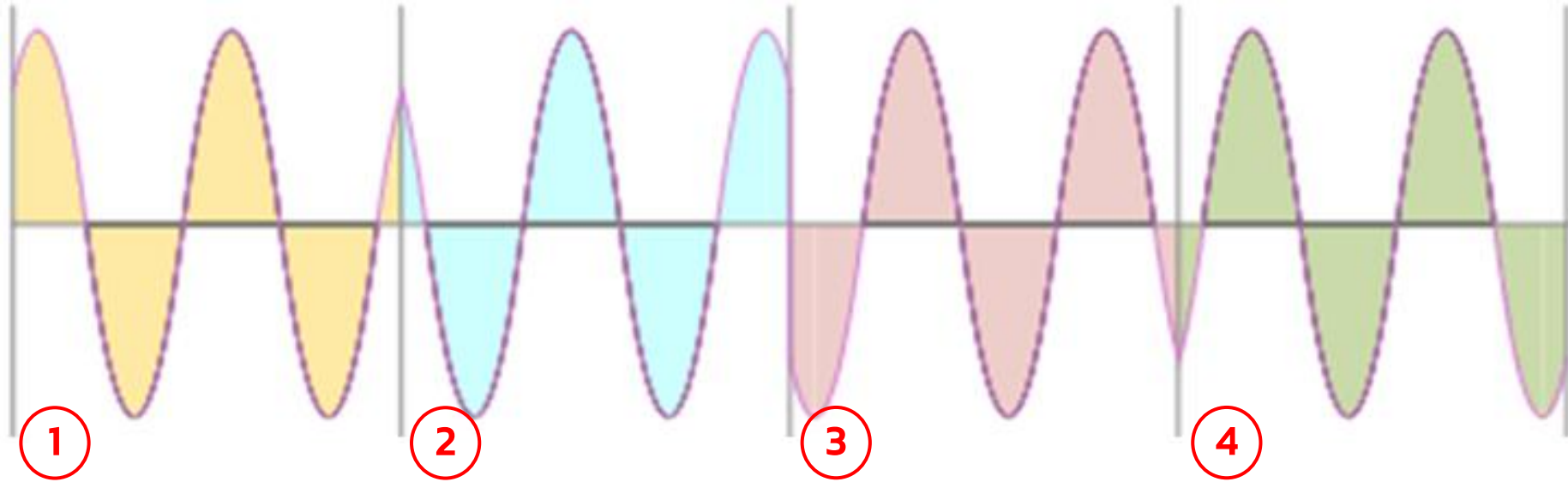


แต่ละ subcarrier จะอยู่ชิด ๆ กันได้



OFDM Signal Frequency Spectra

ถ้าใช้ QPSK แล้วทำ Forward DFT จะรู้ได้อย่างไรว่าที่ส่งมาเป็นแบบ 1, 2, 3 หรือ 4 เพราะ DFT แล้วจะได้ amp. หรือปริมาณของ cosine และ sine เท่านั้น



ต้องทำ DFT 4 ครั้ง ของช่วงที่ 1, 2, 3, 4 แยกกัน

ดูเฉพาะความถี่ที่สนใจ ใช้สูตร $\sin(A + B) = \sin(A)\cos(B) + \cos(A)\sin(B)$

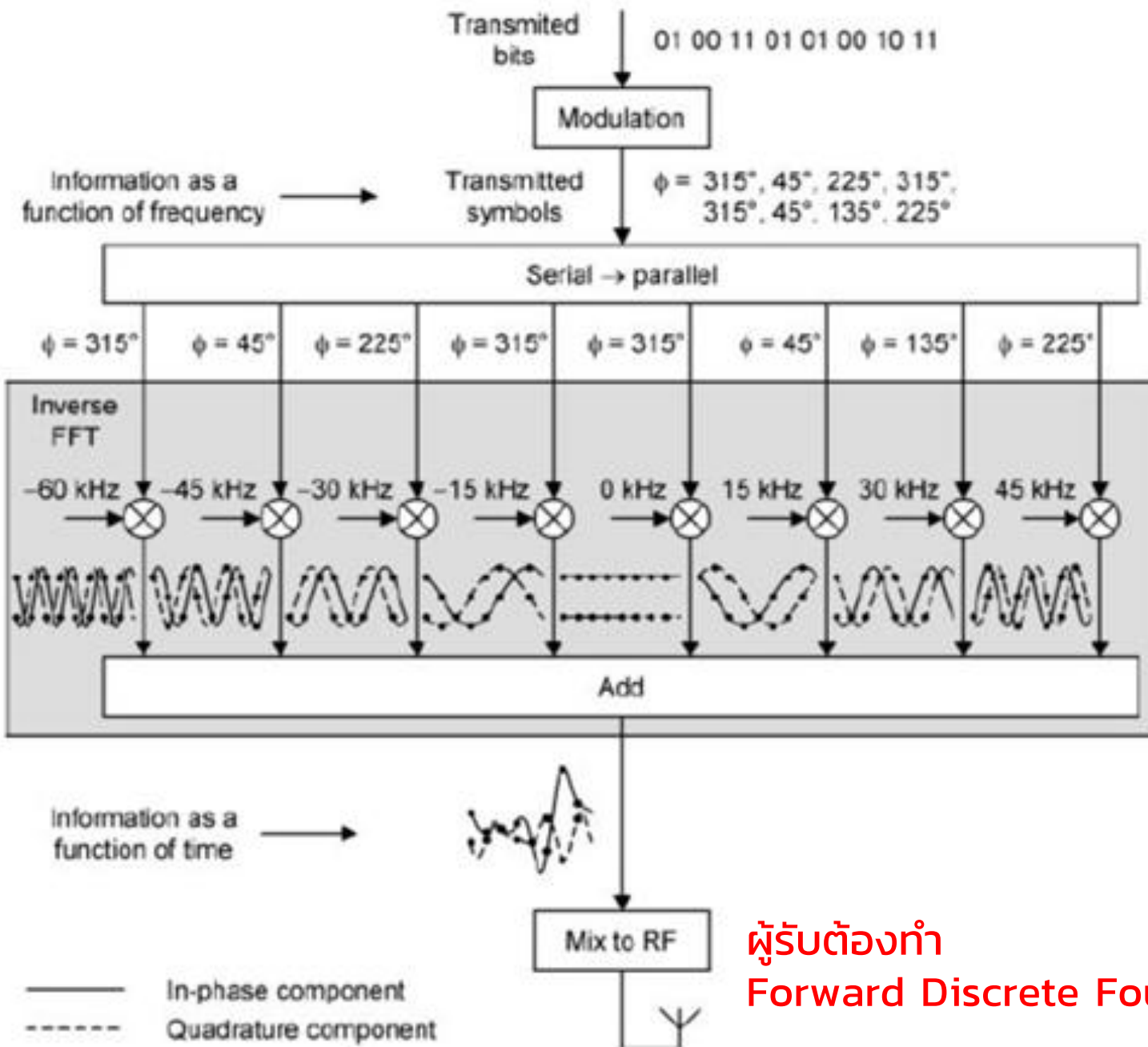
สาระสำคัญของ OFDM อยู่ที่ Forward DFT
ถ้าแยกดูแต่ละความถี่ได้ เห็น +/- ก็จบ

กรณี 1: $f(x) = \sin(x + 45) = \sin(x)\cos(45) + \cos(x)\sin(45) = \frac{1}{\sqrt{2}} \sin(x) + \frac{1}{\sqrt{2}} \cos(x) = (+, +)$

กรณี 2: $f(x) = \sin(x + 135) = \sin(x)\cos(135) + \cos(x)\sin(135) = -\frac{1}{\sqrt{2}} \sin(x) + \frac{1}{\sqrt{2}} \cos(x) = (-, +)$

กรณี 3: $f(x) = \sin(x + 225) = \sin(x)\cos(225) + \cos(x)\sin(225) = -\frac{1}{\sqrt{2}} \sin(x) - \frac{1}{\sqrt{2}} \cos(x) = (-, -)$

กรณี 4: $f(x) = \sin(x + 315) = \sin(x)\cos(315) + \cos(x)\sin(315) = +\frac{1}{\sqrt{2}} \sin(x) - \frac{1}{\sqrt{2}} \cos(x) = (+, -)$

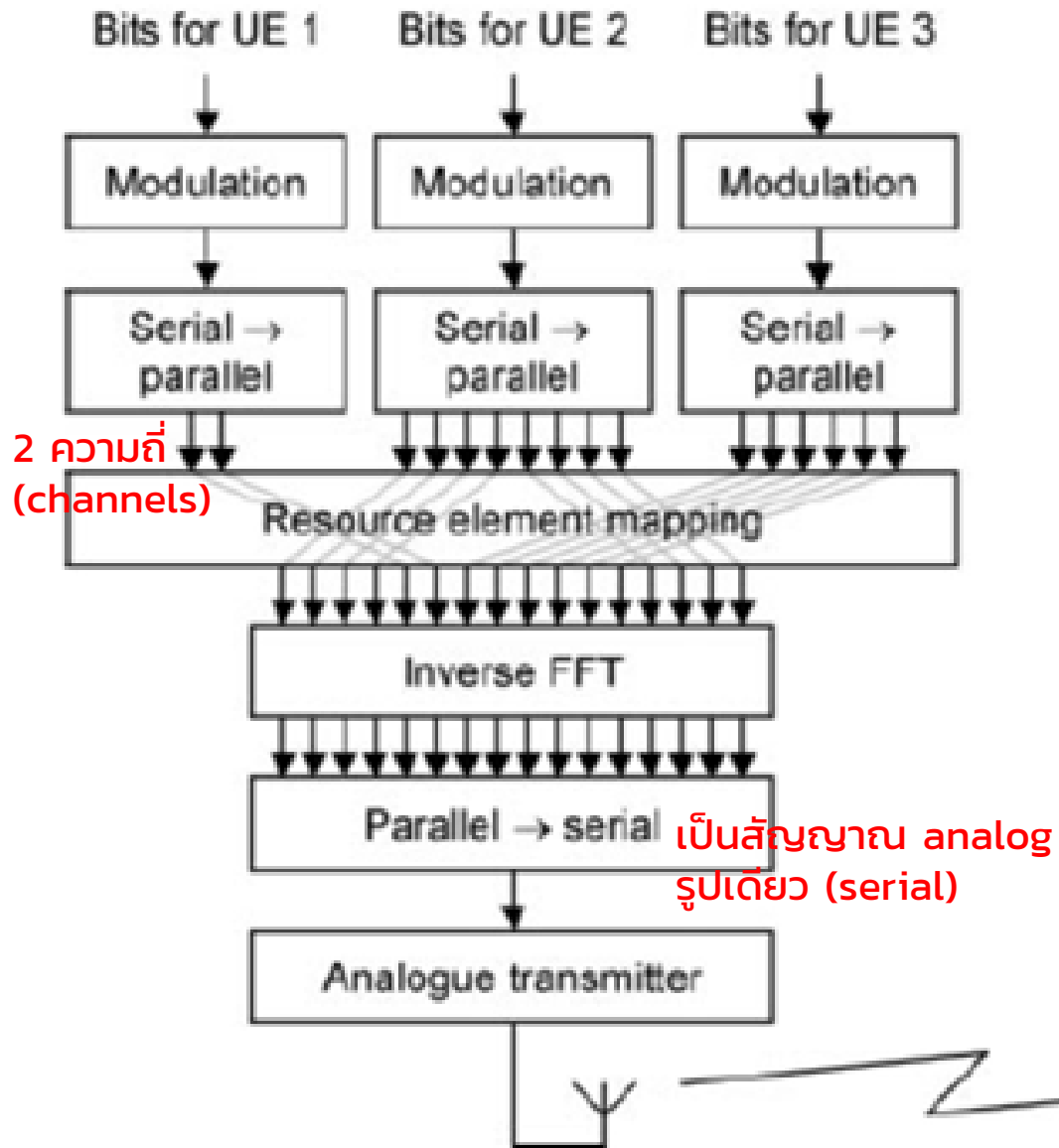


การทำ digital modulation โดยใช้
Quadrature phase shift keying (QPSK)

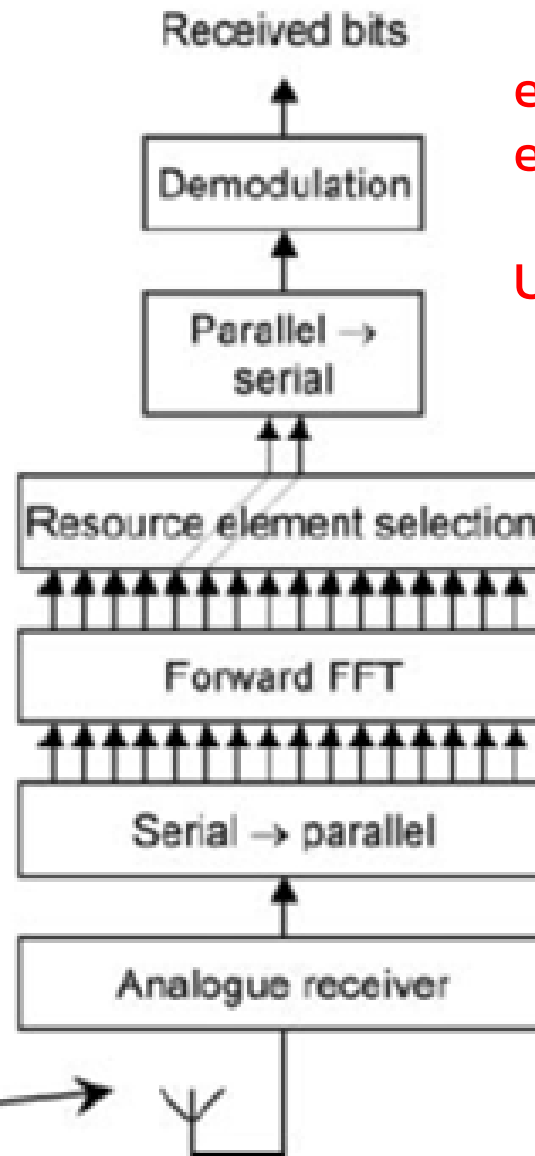
แต่ละความถี่ต้องตั้งฉากกัน (orthogonal)
ผู้ส่งทำ Inverse DFT ไม่สร้าง cosine/sine
wave ของแต่ละความถี่ แล้วผสมกันในอากาศ
ซึ่งไม่มีประสิทธิภาพ เพราะต้องใช้เสาอากาศ
หลายตัว

ผู้รับต้องทำ
Forward Discrete Fourier Transform (Forward DFT)

eNB transmitter



UE 1 receiver

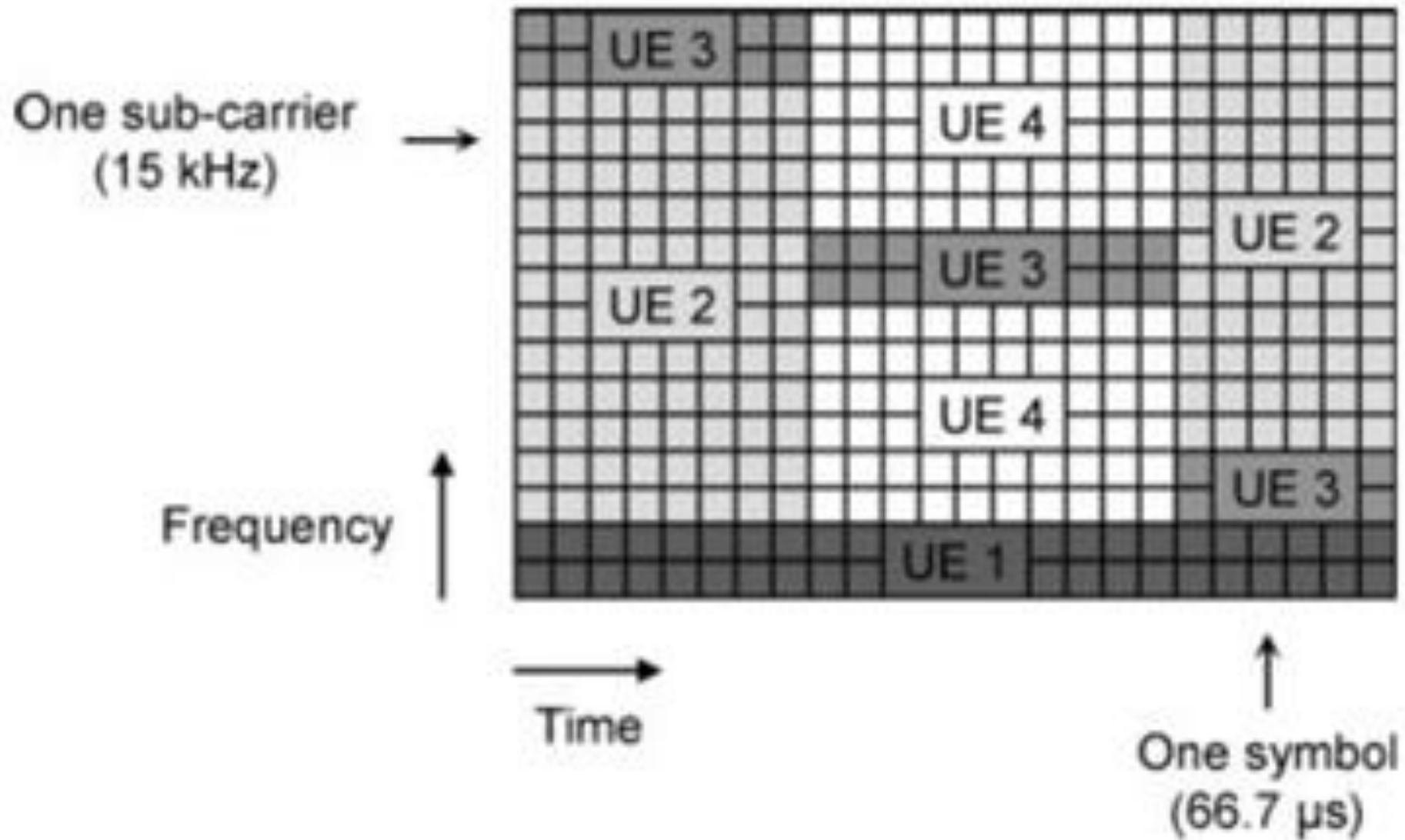


4G LTE (Long Term Evolution)

eNB = Evolved Node B

eNB คือ Base Transceiver Station

UE = User Equipment



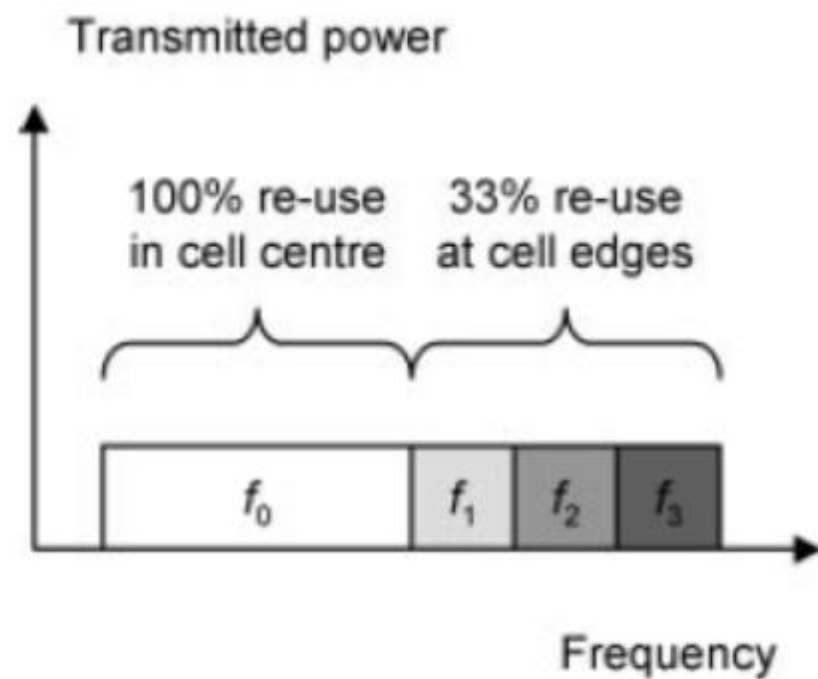
การจัดการความถี่ของ OFDM ที่ Base Transceiver Station (BTS)

Why 15 KHz for sub-carrier spacing ?

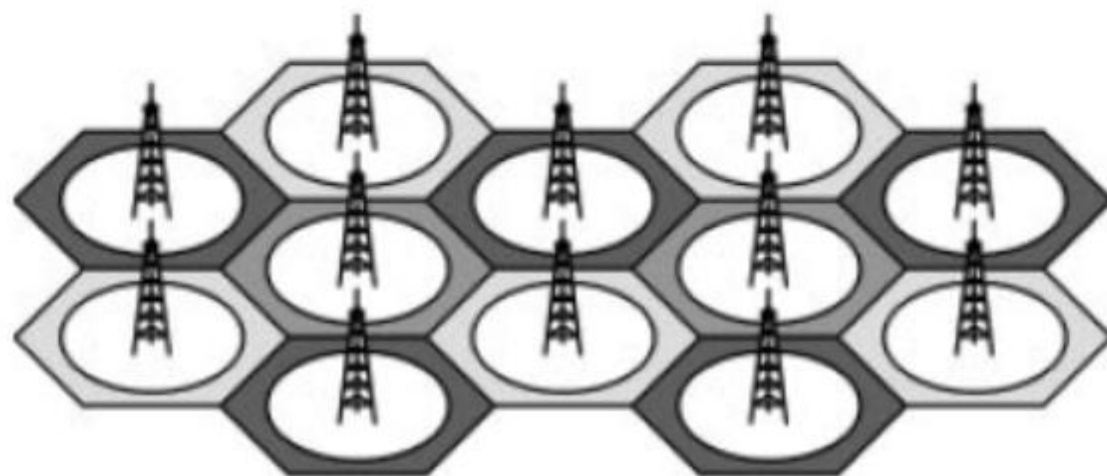
We can draw the following conclusions. If the sub-carrier spacing were much less than 15 kHz, then the system would be prone to interference between the sub-carriers at high mobile speeds. If it were much greater, then the system would be prone to inter symbol interference in large, cluttered cells. The chosen sub-carrier spacing is the result of a trade-off between these two extremes.

ถ้าลด sub-carrier spacing แล้ว จะเสี่ยงต่อการรบกวนเมื่อเคลื่อนที่เร็ว ๆ เช่น ขับรถ ขึ้นรถไฟความเร็วสูง
ถ้าเพิ่ม sub-carrier spacing แล้ว จะเสี่ยงต่อการเกิด inter-symbol interference (ISI)
เพื่อให้ data rate เท่าเดิม ก็ต้องใช้ symbol duration ที่สั้นลง

ความถี่ที่ใช้ส่งข้อมูล (random data + QPSK) เป็นตัวกำหนดว่า sub-carrier spacing จะต้องเป็นเท่าใด
ถ้าใช้ความถี่ในช่วง 1 GHz (mobile comm.) จะได้ sub-carrier spacing เท่ากับ 15 KHz
ถ้าเปลี่ยนความถี่ไป sub-carrier spacing ก็จะเปลี่ยนตาม



(a)

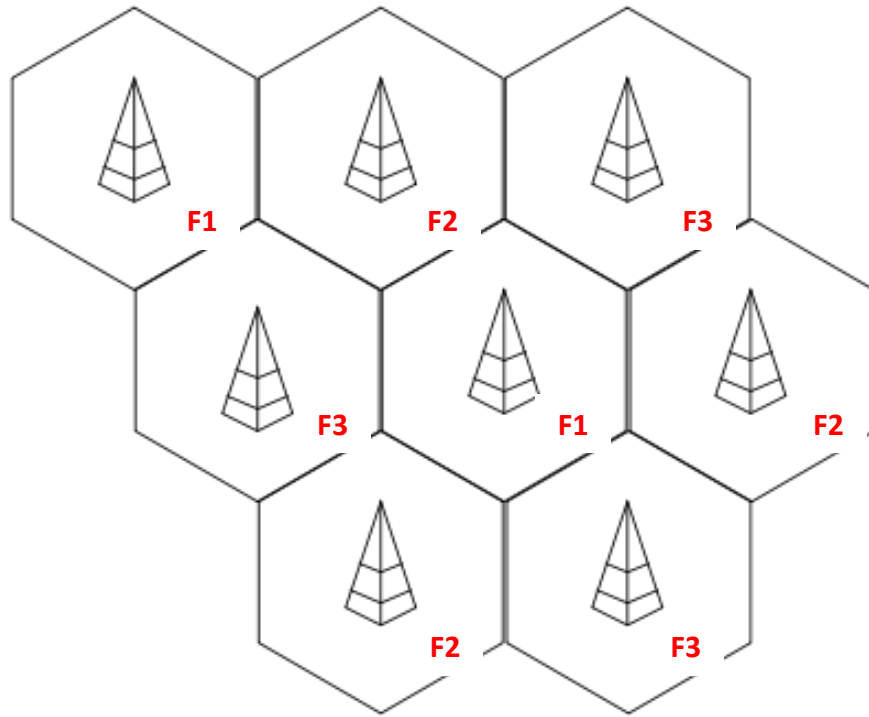


(b)

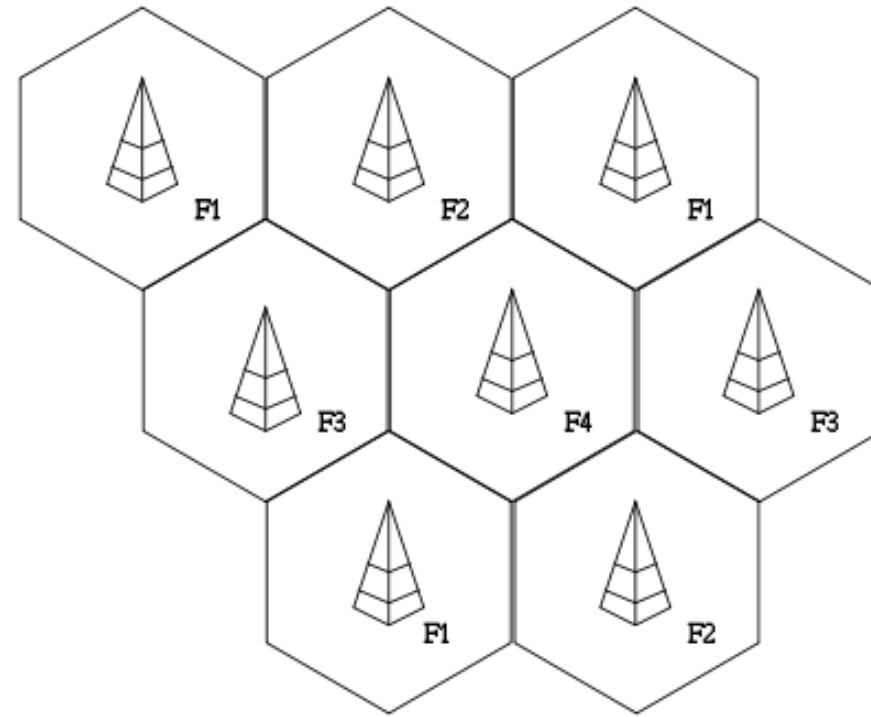
Figure 4.6 Example implementation of fractional frequency re-use when using OFDMA. (a) Use of the frequency domain. (b) Resulting network plan.

- GSM (2G) มี re-use factor ได้สูงสุดแค่ $1/3$ (33.33%)
- CDMA (3G) มี re-use factor 100% มีการรบกวนสูงที่รอยต่อระหว่าง cell (ทุก cell ใช้ความถี่เหมือนกันหมด)
- OFDMA (4G)
 - Cell center: re-use factor 100%
 - Cell edge: re-use factor 33.33%

ใช้ความถี่ที่ใกล้เคียงกันมาก แต่ไม่รบกวนกัน

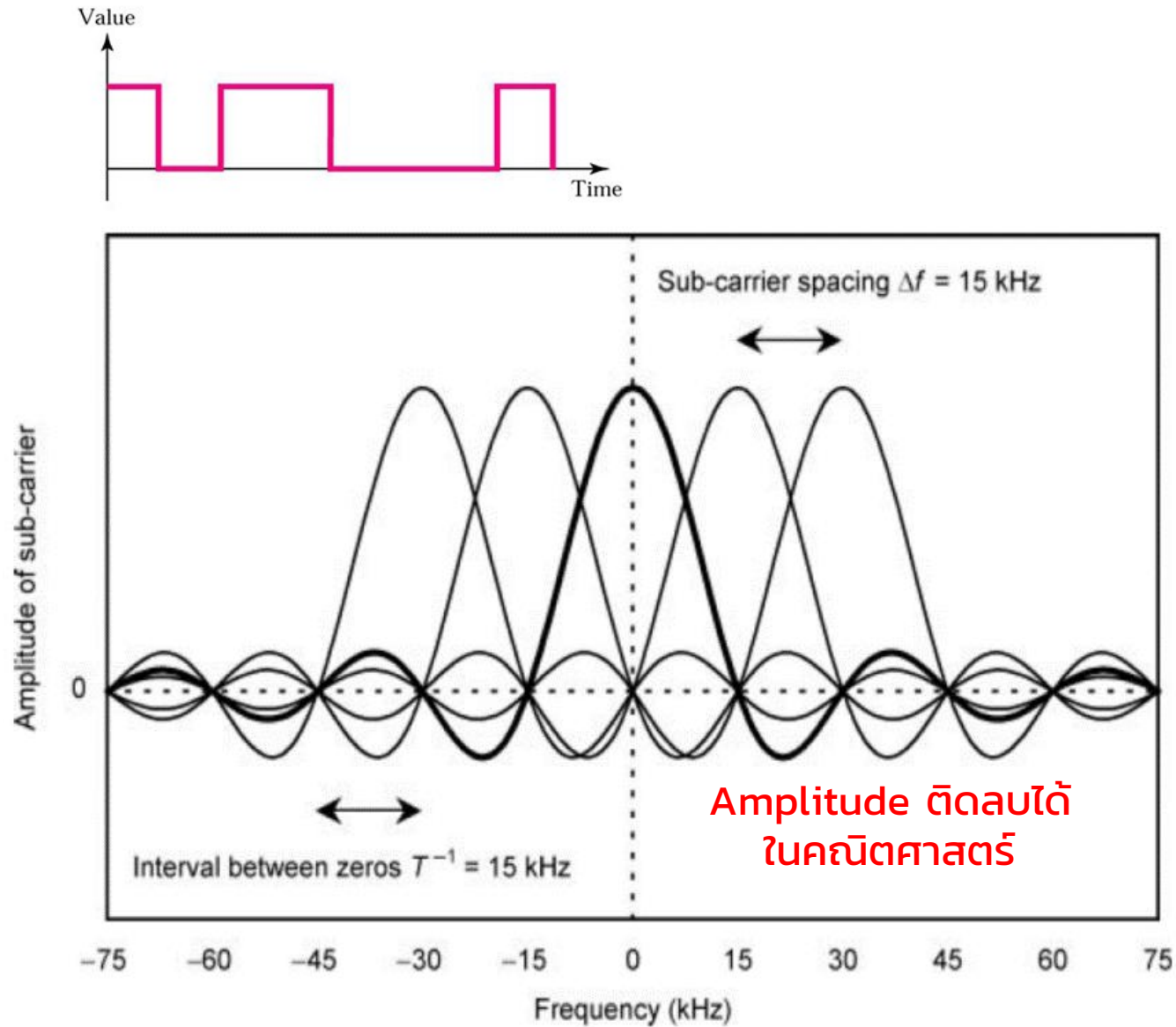


re-use factor = $1/3$ (33.33%)

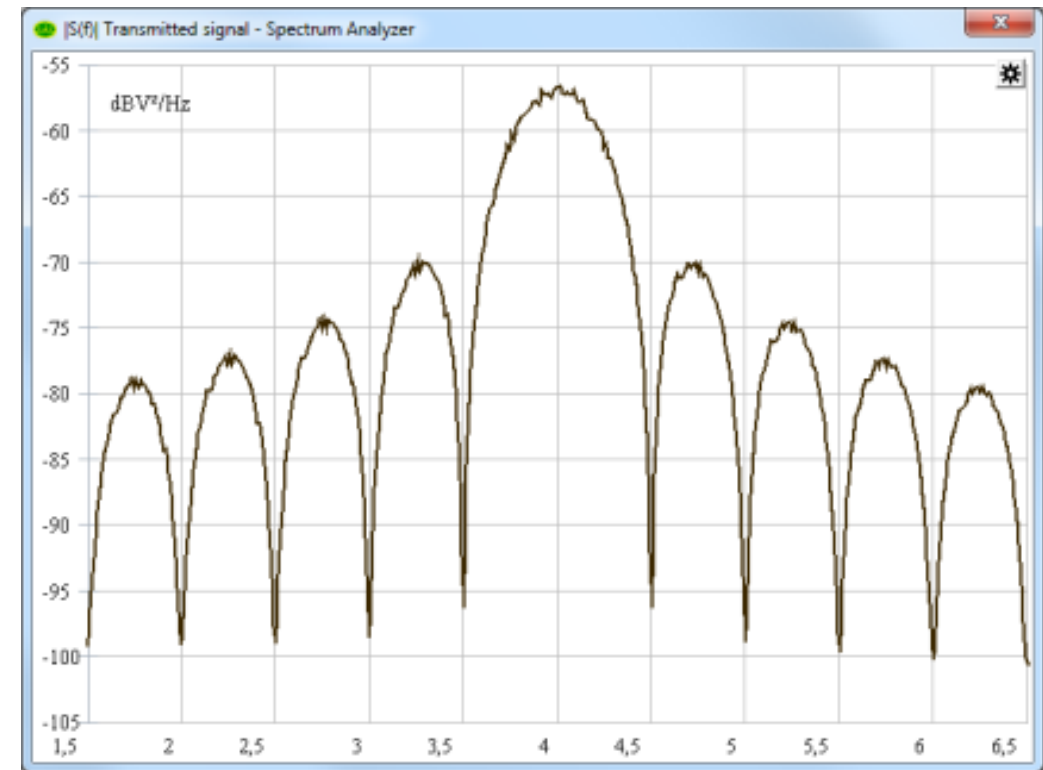


re-use factor = $1/4$ (25%)

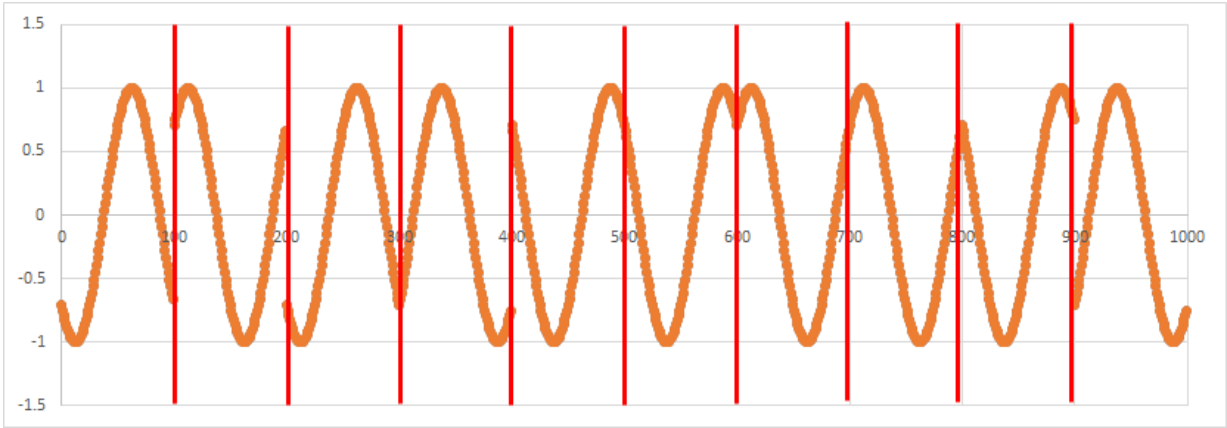
ถ้าเราจะอธิบาย OFDM ด้วยรูปนี้ มันสร้างจาก square wave ซึ่งเป็นการส่งข้อมูลแบบใช้สาย (wired) บนอากาศ/อวกาศ (wireless) ใช้ square wave ไม่ได้



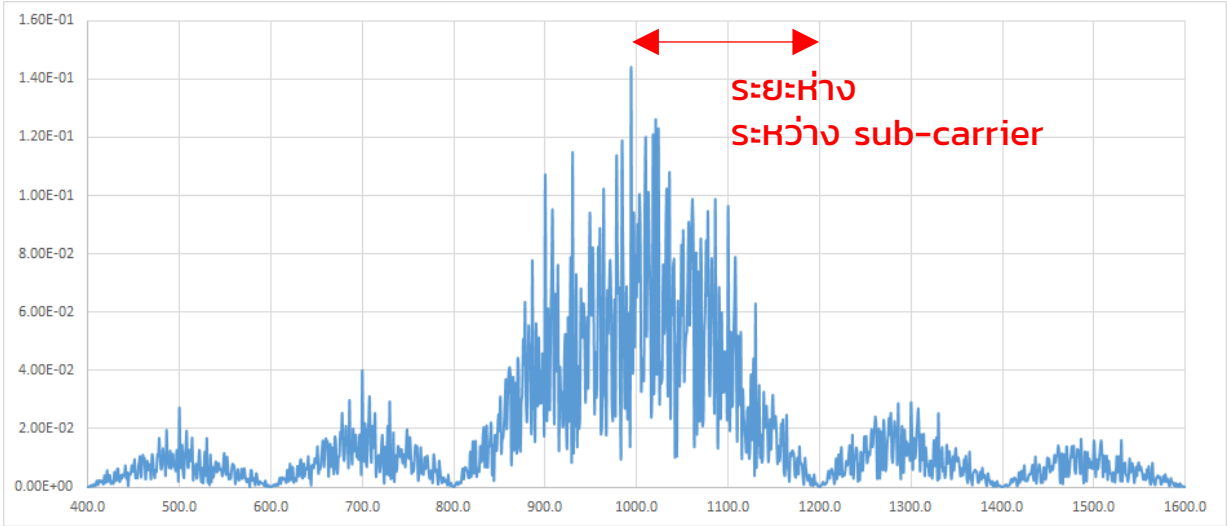
ต้องใช้ QPSK ถึงจะได้รูปนี้ และรูปนี้แกน y เป็น log scale



การคำนวณหาระยะห่างระหว่าง sub-carrier

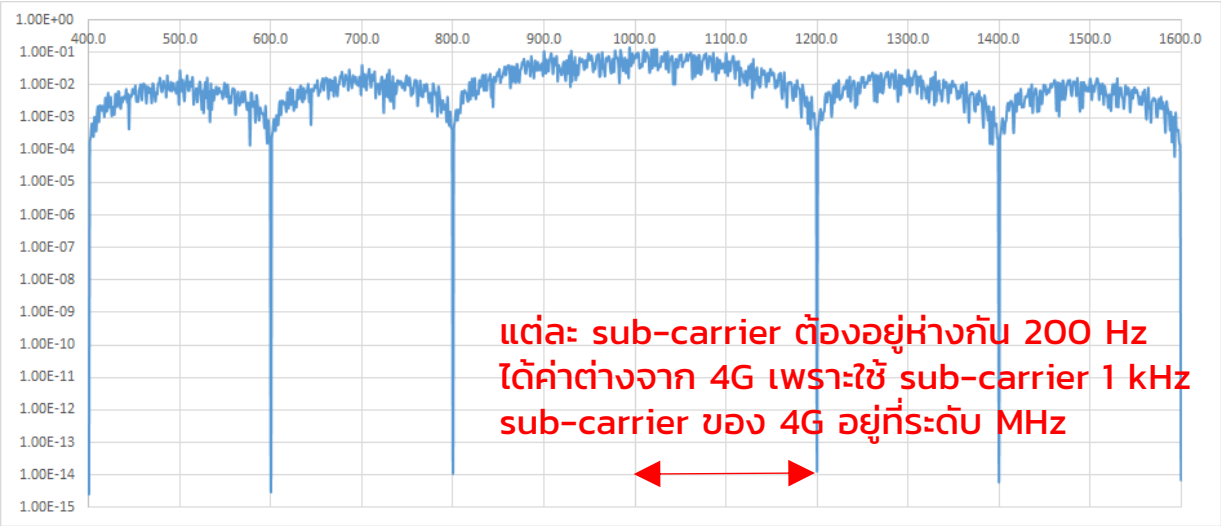


Sub-carrier	1,000	Hz	
Sampling rate	10,000	Hz	
N	10,000	points	
phase change	50	points	(200 symbols per second)



$\cos^2 + \sin^2$

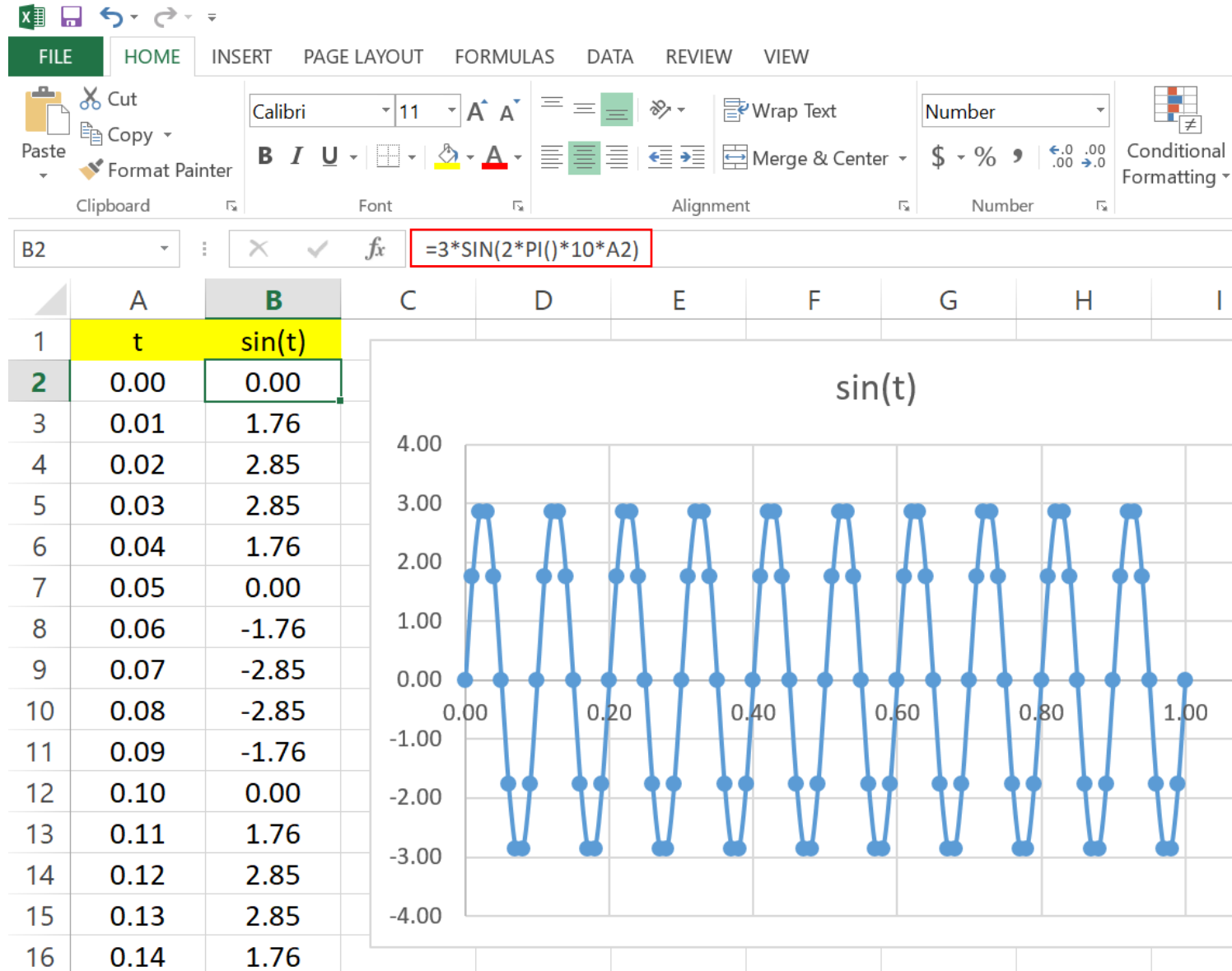
(พลังงานรวม ที่แต่ละความถี่)



Log scale

แต่ละ sub-carrier ต้องอยู่ห่างกัน 200 Hz
ได้ค่าต่างจาก 4G เพราะใช้ sub-carrier 1 kHz
sub-carrier ของ 4G อยู่ที่ระดับ MHz

Plot sine/cosine in Excel



$$y = a \cdot \sin(2\pi \cdot f \cdot t)$$

a คือ amplitude

f คือ frequency (Hz)

ในรูป $a = 3$ และ $f = 10 \text{ Hz}$

```

1  using System;
2  using System.IO;
3
4  public class MyProg {
5
6      public static void Main(string[] Args) {
7
8          if (Args.Length != 3) {
9              Console.WriteLine("ForwardDFT.exe [input_file] [sampling_rate] [output_file]");
10             return;
11         }
12
13         string input_file = Args[0];
14         double sampling_rate = Convert.ToDouble(Args[1]);
15         string output_file = Args[2];
16
17         int N = 0;
18
19         StreamReader SR = new StreamReader(input_file);
20         while (SR.ReadLine() != null) N++;
21         SR.Close();
22
23         if (N % 2 != 0) {
24             Console.WriteLine("N must be an even number.");
25             return;
26         }
27
28         double[] x = new double[N];
29
30         SR = new StreamReader(input_file);
31         for (int i = 0; i < N; i++) {
32             x[i] = Convert.ToDouble(SR.ReadLine());
33         }
34         SR.Close();
35     }
36 }
37

```

S: input
 เป็นชื่อ text file
 sampling
 rate
 S: output
 เป็นชื่อ text file

```
39 // page 158: The Scientist and Engineer's Guide to Digital Signal Processing
```

```
40  
41 double[] ReX = new double[N/2+1];
```

```
42 double[] ImX = new double[N/2+1];
```

```
43  
44 for (int k = 0; k < N/2+1; k++) {
```

```
45     ReX[k] = ImX[k] = 0;
```

```
46     for (int i = 0; i < N; i++) {
```

```
47         ReX[k] += x[i] * Math.Cos((double)2 * Math.PI * (double)k * (double)i / (double)N);
```

```
48         ImX[k] += -x[i] * Math.Sin((double)2 * Math.PI * (double)k * (double)i / (double)N);
```

```
49     }
```

```
50 }
```

```
51  
52 // page 153: The Scientist and Engineer's Guide to Digital Signal Processing
```

```
53  
54 double[] ReX_ = new double[N/2+1];
```

```
55 double[] ImX_ = new double[N/2+1];
```

```
56  
57 for (int k = 0; k < N/2+1; k++) {
```

```
58     ReX_[k] = ReX[k] / (double) (N/2);
```

```
59     ImX_[k] = -ImX[k] / (double) (N/2);
```

```
60  
61     // exception cases
```

```
62     if (k == 0) ReX_[0] = ReX[0] / (double)N;
```

```
63     if (k == N/2) ReX_[N/2] = ReX[N/2] / (double)N;
```

```
64 }
```

```
65  
66  
67 StreamWriter SW = new StreamWriter(output_file);
```

```
68  
69 SW.WriteLine("k\tReX_[k]\tImX_[k]\tfrequency");
```

```
70 for (int k = 0; k < N/2+1; k++) {
```

```
71     double fraction = (double)k / (double)N;
```

```
72     SW.WriteLine(k + "\t" + ReX_[k] + "\t" + ImX_[k] + "\t" + String.Format("{0:0.000000}", fraction * sampling_rate));
```

```
73 }
```

```
74  
75  
76 SW.Close();
```

ไฟล์ input

1	0.00
2	3.42
3	6.59
4	9.31
5	11.40
6	12.72
7	13.20
8	12.85
9	11.72
10	9.95

ใช้จำนวนจุดเป็นเลขคู่
เพราะ N ต้องหารสอง

ถ้ามีจุดเป็นจำนวนคี่
ก็โยนทิ้งไป 1 จุด

Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS

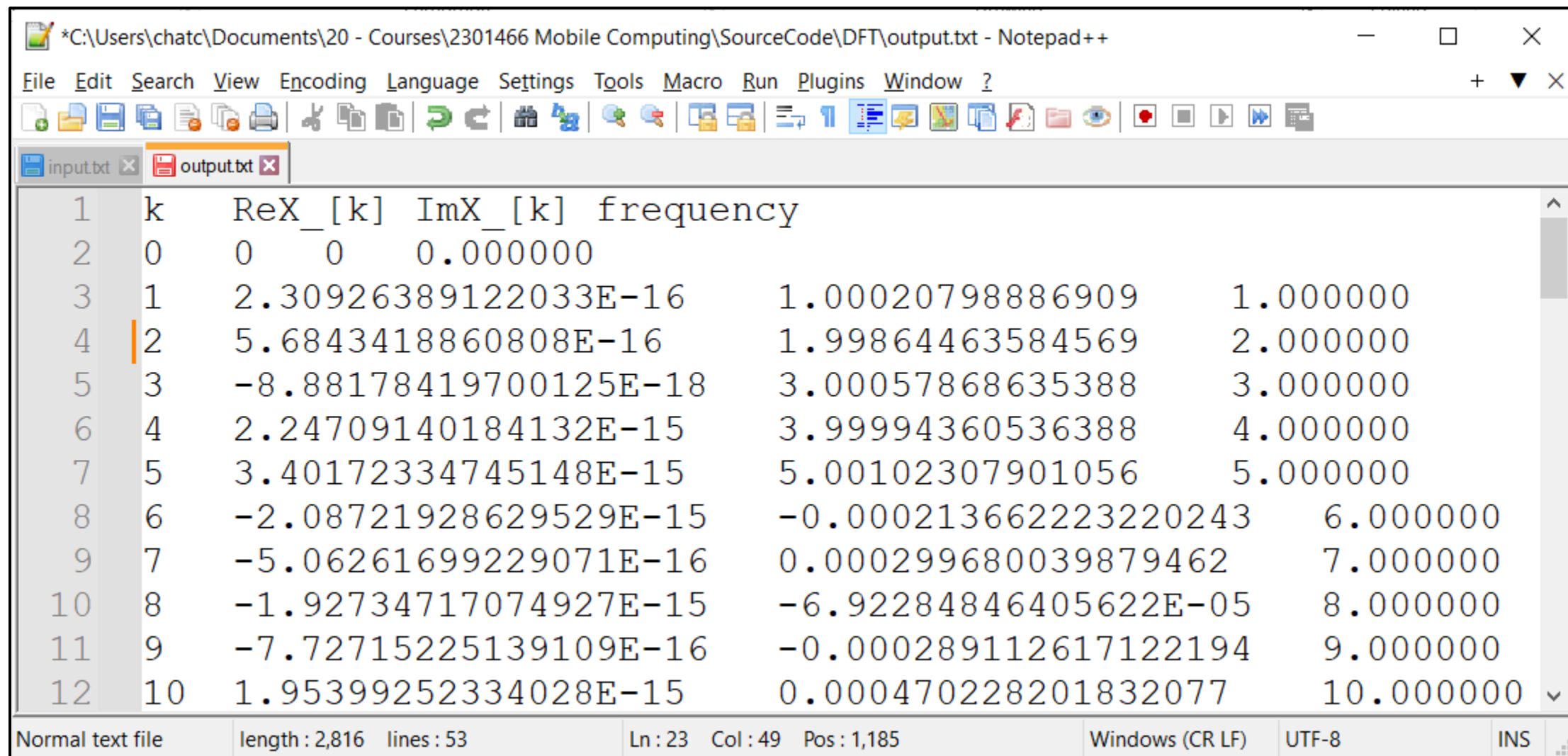
ไฟล์ output

คอลัมน์ 1: ค่า k

คอลัมน์ 2: ปริมาณ cosine (amp.)

คอลัมน์ 3: ปริมาณ sine (amp.)

คอลัมน์ 4: ความถี่ (Hz)



	k	ReX_[k]	ImX_[k]	frequency
1	0	0	0	0.000000
2	1	2.30926389122033E-16	1.00020798886909	1.000000
3	2	5.6843418860808E-16	1.99864463584569	2.000000
4	3	-8.88178419700125E-18	3.00057868635388	3.000000
5	4	2.24709140184132E-15	3.99994360536388	4.000000
6	5	3.40172334745148E-15	5.00102307901056	5.000000
7	6	-2.08721928629529E-15	-0.000213662223220243	6.000000
8	7	-5.06261699229071E-16	0.000299680039879462	7.000000
9	8	-1.92734717074927E-15	-6.92284846405622E-05	8.000000
10	9	-7.72715225139109E-16	-0.000289112617122194	9.000000
11	10	1.95399252334028E-15	0.000470228201832077	10.000000

Expo

`npx create-expo-app@latest --template blank` ตั้งชื่อแอปว่า to-do

ติดตั้ง Android studio <https://docs.expo.dev/get-started/set-up-your-environment/?platform=android&device=simulated>

Android Studio Meerkat | 2024.3.1 Patch 1

More actions -> Virtual Device Manager -> Medium Phone API 36 -> Start

`npm install @react-navigation/native`

`npm install @react-navigation/stack`

JS App.js X

JS App.js > App

```
1  import { StatusBar } from 'expo-status-bar';
2  import { StyleSheet, Text, View } from 'react-native';
3
4  export default function App() {
5    return (
6      <View style={styles.container}>
7        <Text>Hello world! Chatchawit</Text>
8        <StatusBar style="auto" />
9      </View>
10   );
11 }
12
13 const styles = StyleSheet.create({
14   container: {
15     flex: 1,
16     backgroundColor: '#fff',
17     alignItems: 'center',
18     justifyContent: 'center',
19   },
20 });
```

screens > JS SignIn.js > ...

```
1  import React from "react";
2  import { View, Text, Button, StyleSheet } from "react-native";
3  import { useNavigation } from "@react-navigation/native";
4
5  const SignIn = () => {
6    const navigation = useNavigation();
7
8    return (
9      <View style={styles.container}>
10        <Text style={styles.title}>SignIn</Text>
11        <Button title="Go to Main" onPress={() => navigation.navigate("Main")} />
12        <Button title="Go to Credit" onPress={() => navigation.navigate("Credit")} />
13      </View>
14    );
15  };
16
17  const styles = StyleSheet.create({
18    container: {
19      flex: 1,
20      justifyContent: "center",
21      alignItems: "center",
22      padding: 20,
23    },
24    title: {
25      fontSize: 24,
26      marginBottom: 20,
27    },
28  });
29
30  export default SignIn;
```

screens > JS Main.js > ...

```
1  import React from 'react';
2  import { View, Text, Button, StyleSheet } from 'react-native';
3  import { useNavigation } from '@react-navigation/native';
4
5  const Main = () => {
6    const navigation = useNavigation();
7
8    return (
9      <View style={styles.container}>
10        <Text style={styles.title}>Main Screen</Text>
11        <Button
12          title="Back to Home"
13          onPress={() => navigation.goBack()}
14        />
15      </View>
16    );
17  };
18
19  const styles = StyleSheet.create({
20    container: {
21      flex: 1,
22      justifyContent: 'center',
23      alignItems: 'center',
24    },
25    title: {
26      fontSize: 24,
27      marginBottom: 20,
28    }
29  });
30
31  export default Main;
```

screens > JS Credit.js > ...

```
1  import React from 'react';
2  import { View, Text, Button, StyleSheet } from 'react-native';
3  import { useNavigation } from '@react-navigation/native';
4
5  const Credit = () => {
6    const navigation = useNavigation();
7
8    return (
9      <View style={styles.container}>
10        <Text style={styles.title}>Credit Screen</Text>
11        <Button
12          title="Back to Home"
13          onPress={() => navigation.goBack()}
14        />
15      </View>
16    );
17  };
18
19  const styles = StyleSheet.create({
20    container: {
21      flex: 1,
22      justifyContent: 'center',
23      alignItems: 'center',
24    },
25    title: {
26      fontSize: 24,
27      marginBottom: 20,
28    }
29  });
30
31  export default Credit
```