姓名: 葉冠宏 學號: r11943113

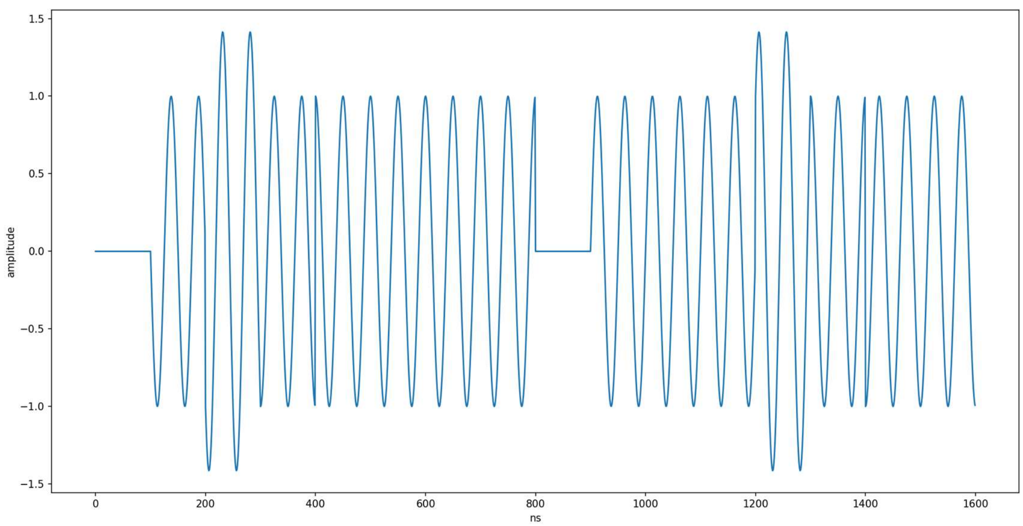
1.

```
[0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1]
[Finished in 145ms]
```

2.

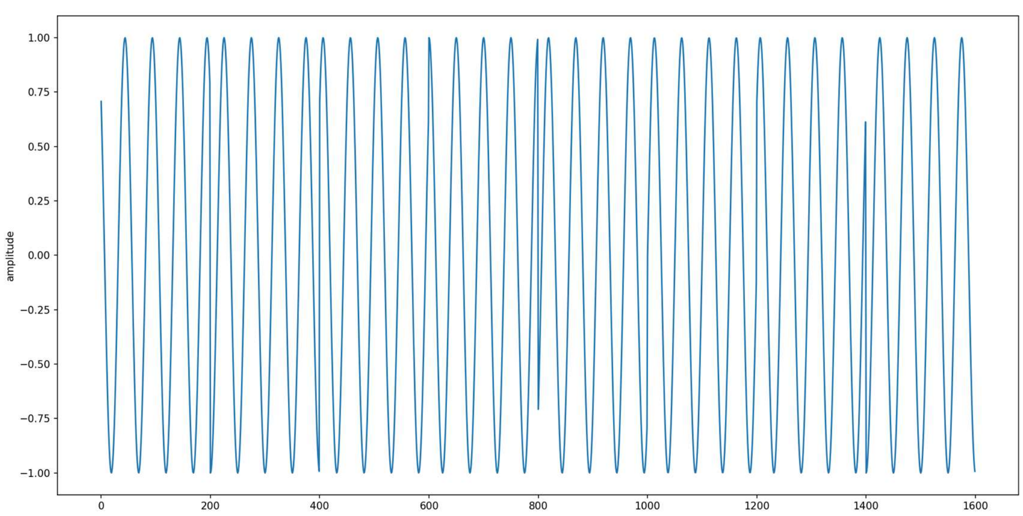
2(a)

請執行 python Q2ac.py



2(b)

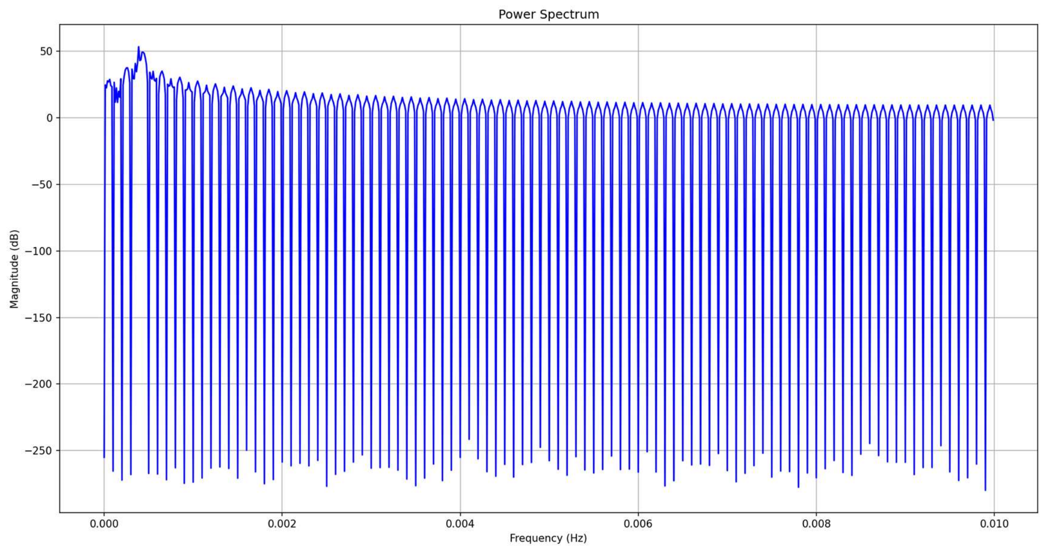
請執行 python Q2bc.py



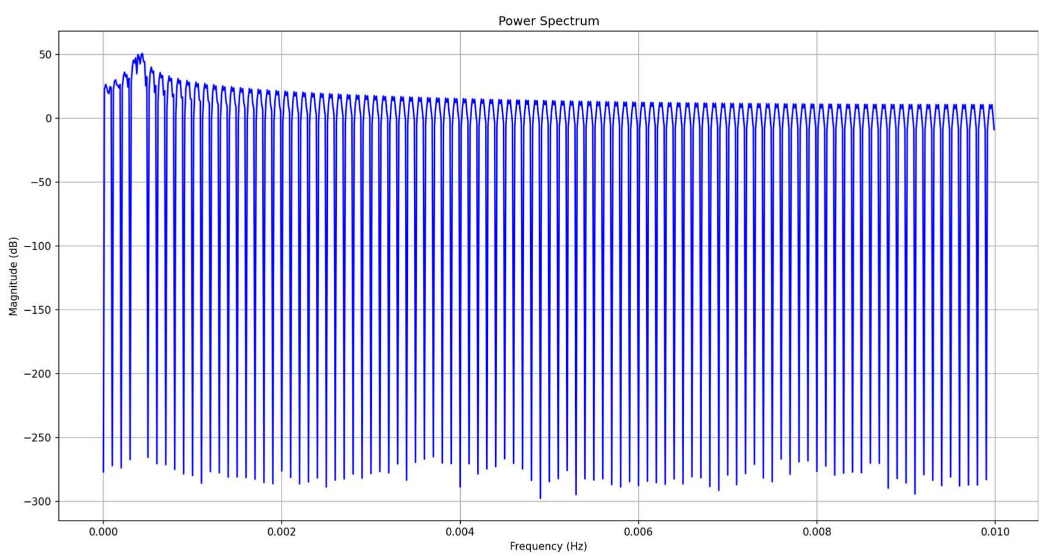
2(c)

請執行 python Q2ac.py 和 python Q2bc.py

Spectrum for O-QPSK:



Spectrum for Pi/4-QPSK:



2(d)

我們可以觀察到，O-QPSK 和 Pi/4-QPSK 在頻譜上主要都集中在靠左的頻率。雖然有加上 phase 相位的改變影響，但因為兩者的 passband 都是用一樣的 carrier frequency f{sub}，因此主要的頻譜長相看起來差不多。然而，Pi/4-QPSK 在主要頻譜和次要頻譜的差距是比較平滑的，而 O-QPSK 的差距則看起來比較大，下降幅度比較明顯。可能原因是因為 Pi/4-QPSK 在相位的轉移上最小可以到 Pi/4，

波動幅度較小。而 O-QPSK 則是在相位轉移上是 pi/2，位移比較大。

3.

3(a)

由題意，我們知道 symbol period 是 500 ns，然後現在有 8 個 subcarrier，所以我們知道 subcarrier spacing fsub 的值就是 symbol period 的倒數。

Fsub=1/(500 * 10^-9) =0.002*10^9 = 2 x 10^6 (Hz) =2 (MHz)

為了去計算 sampling interval Ts，需要符合以下條件:

1. minimum separation to keep orthogonality: fsub = 1/(N * Ts)
2. sampling rate 需要至少是 bandwidth 的兩倍

有 8 個 subcarrier，所以 N=8，subcarrier spacing 是 2 Mhz，所以 total bandwidth 是 8 * 2 Mhz = 16 Mhz。因此 sampling rate 至少要是 32 Mhz。

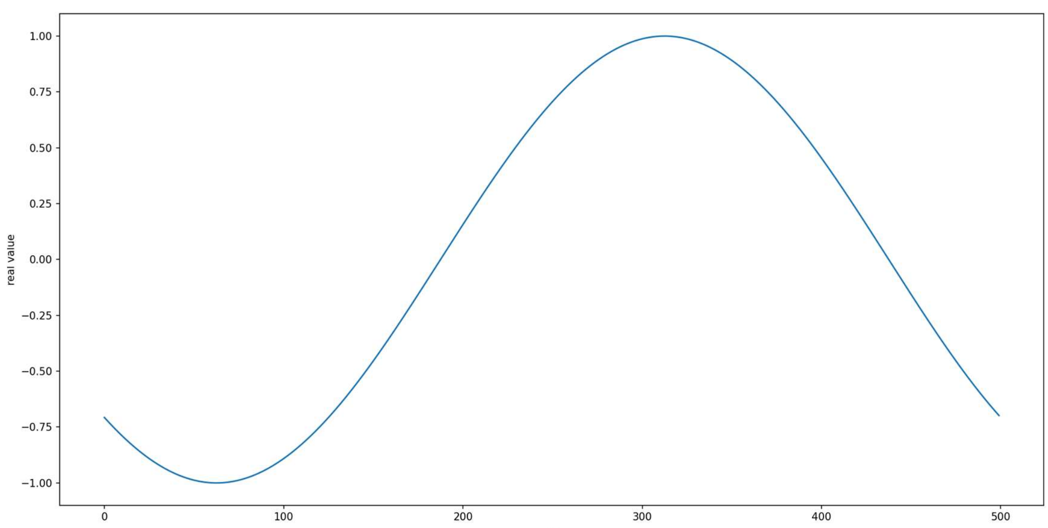
Sampling interval Ts 是 1/(32 * 10^6) = 31.25 ns

3(b)

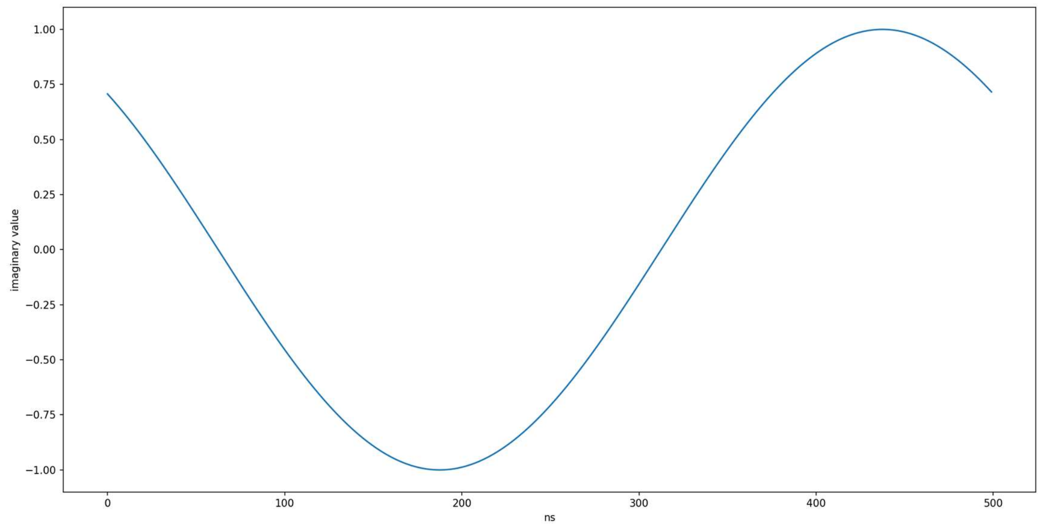
請執行 python Q3b.py

當 k=1 時:

Real part

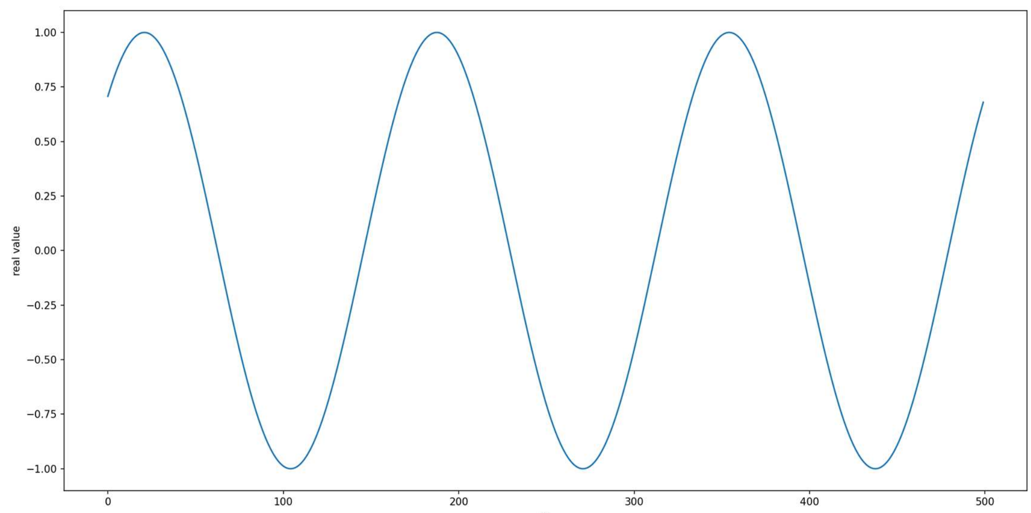


Imaginary part:

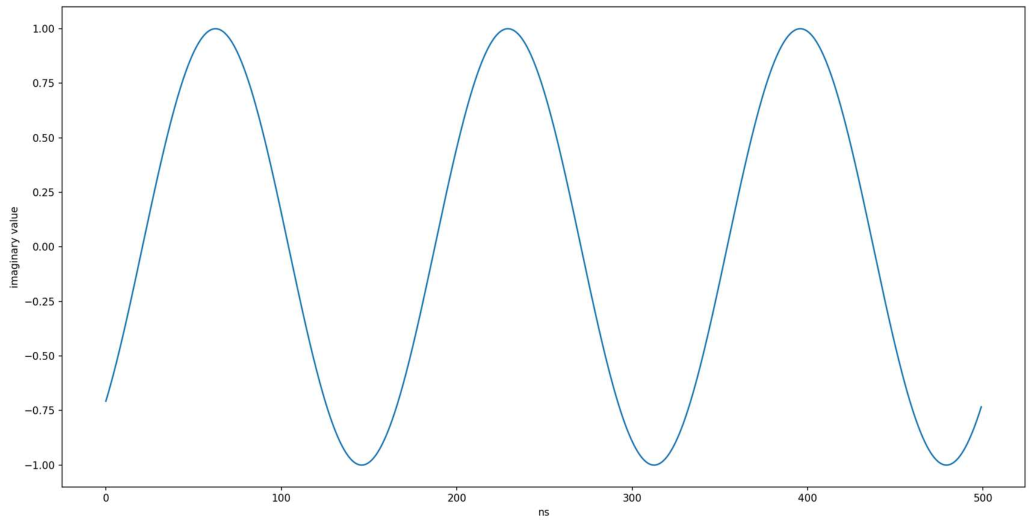


當 k=3 時:

Real part

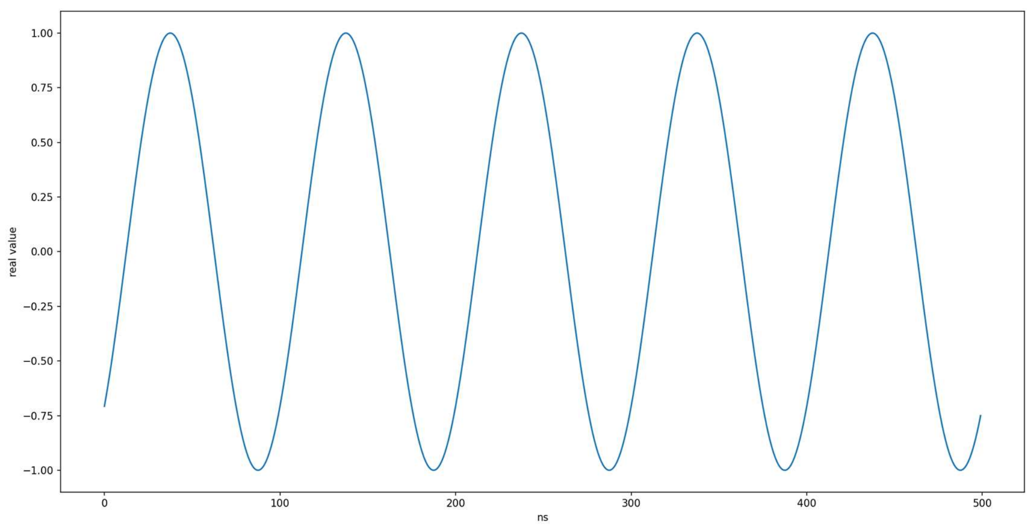


Imaginary part

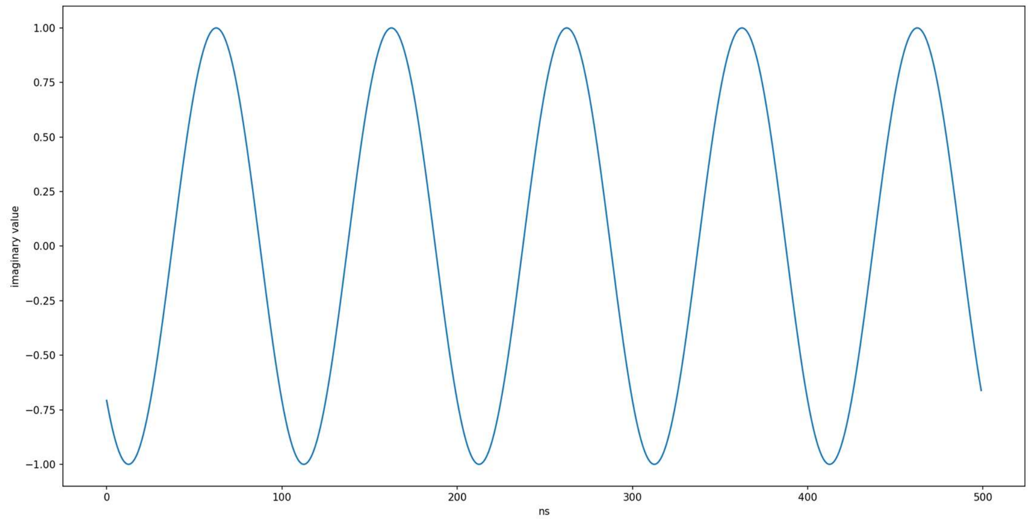


當 k=5 時:

Real part

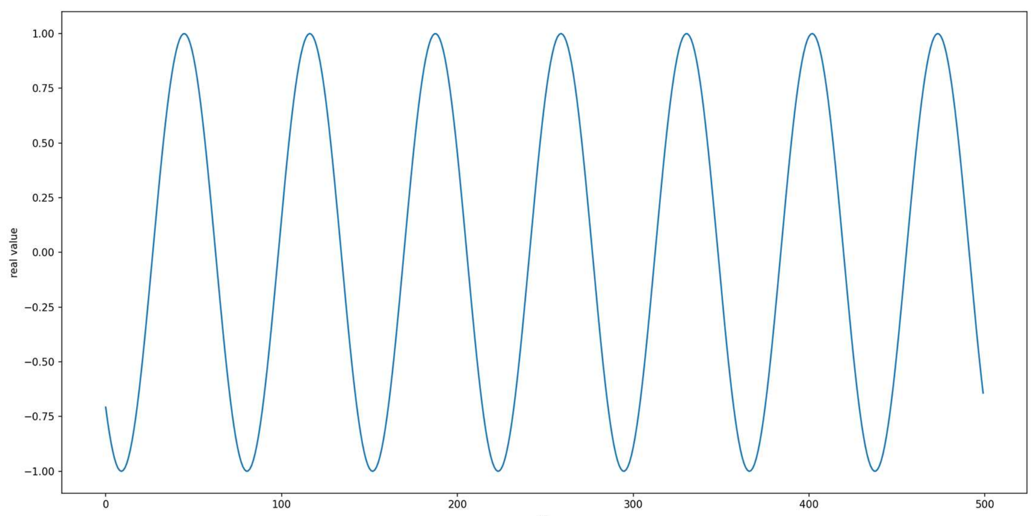


Imaginary part

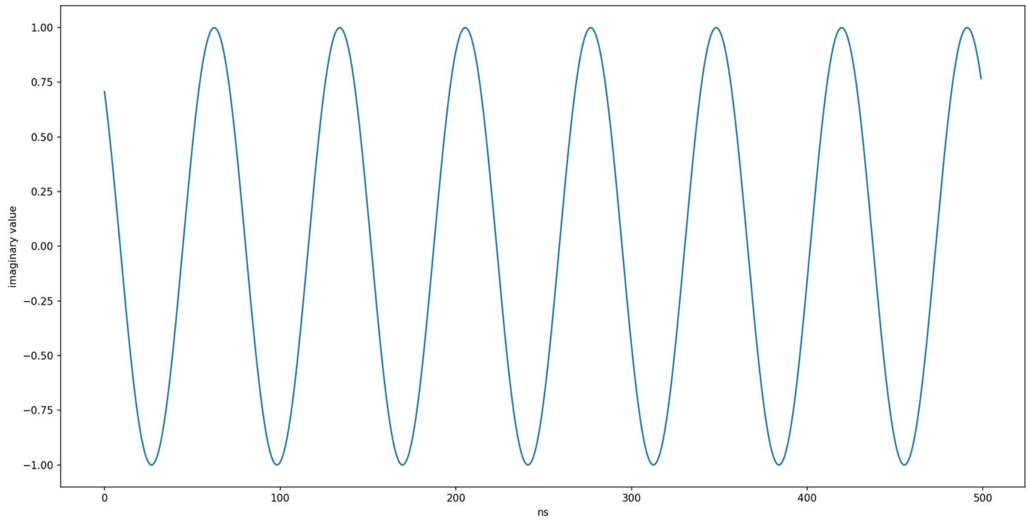


當 k=7 時:

Real part



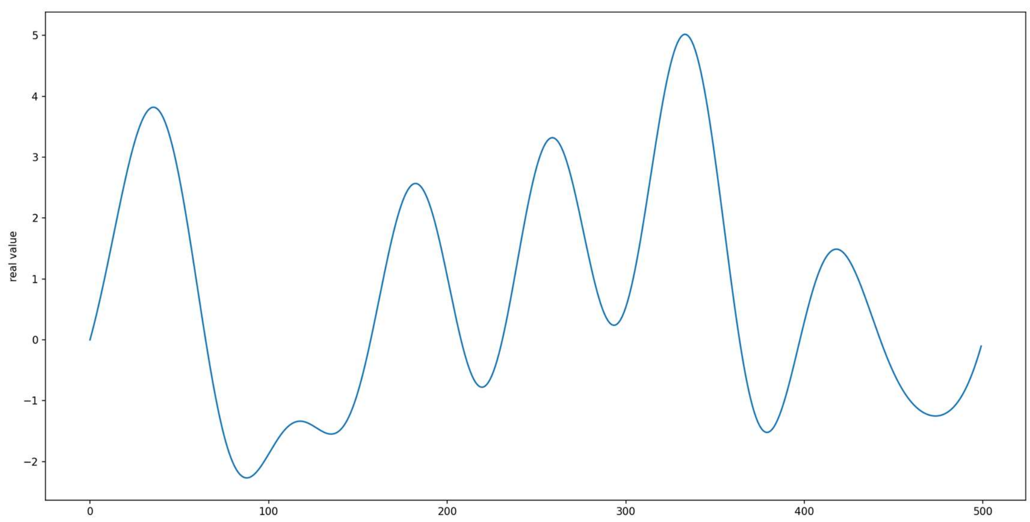
Imaginary part



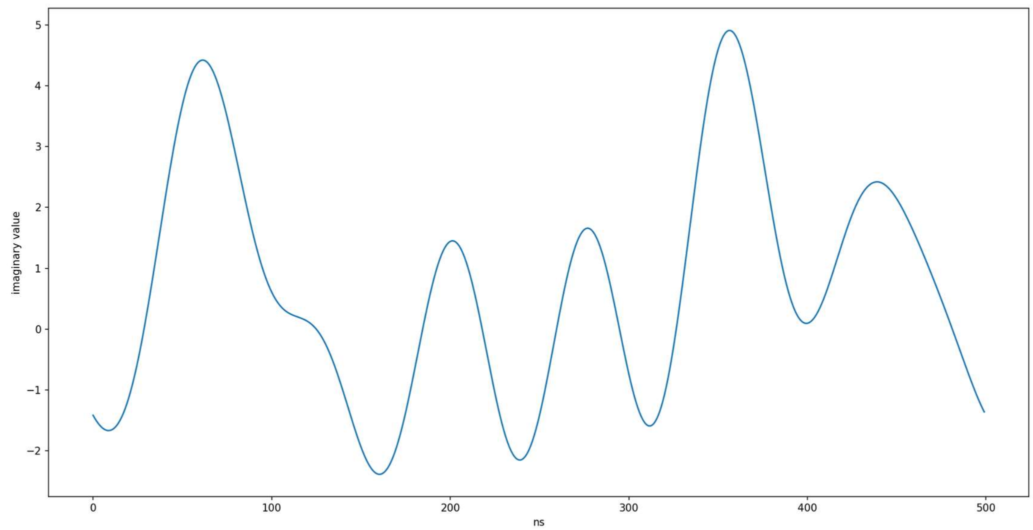
3(c)

請執行 python Q3c.py

Real part



Imaginary part



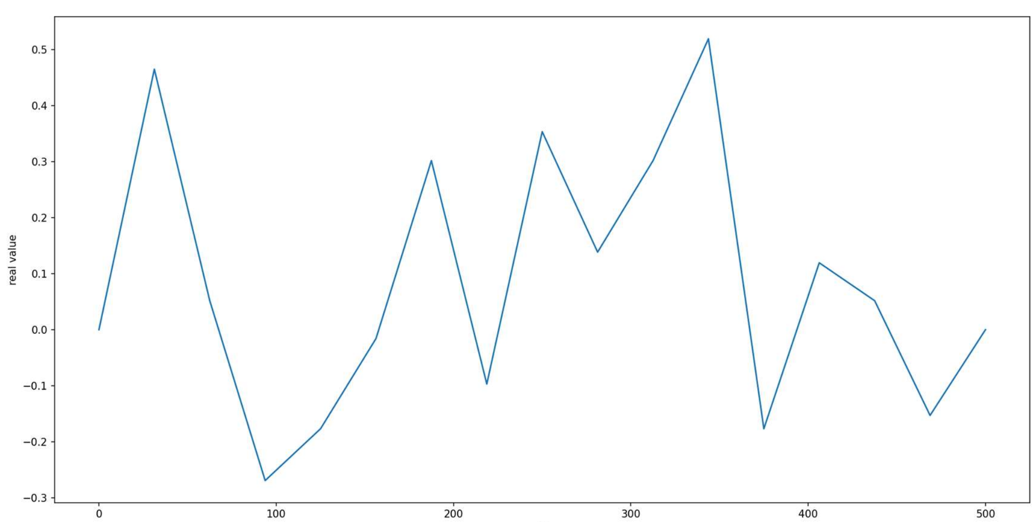
3(d)

請執行 python Q3d.py

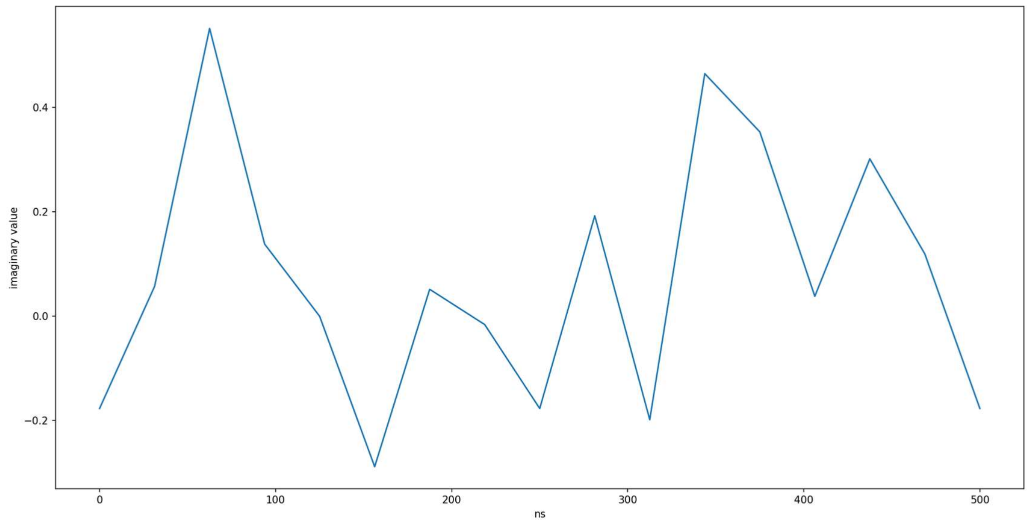
為了要計算某一個時間點 nTs 的 baseband，我們計算 Ts 為 31.25 ns。總共的 n 是 500/31.25，即有 16 個點。n 的值域為[0,16]。fk = k x fsub。fsub 為 2 x 10^-3 (1/ns)。N 為 8，因為有 8 個 subcarrier。Xk 為每個 symbol 的透過 black constellation 的 phase。透過以下的 IFFT 轉換，我們可以計算出每個時間點 n Ts 的 baseband 值是多少。

$$x_{BB}(nT_s) = \sum_{k=0}^{N-1} X_k e^{j2\pi f_k n T_s}$$

Real part:



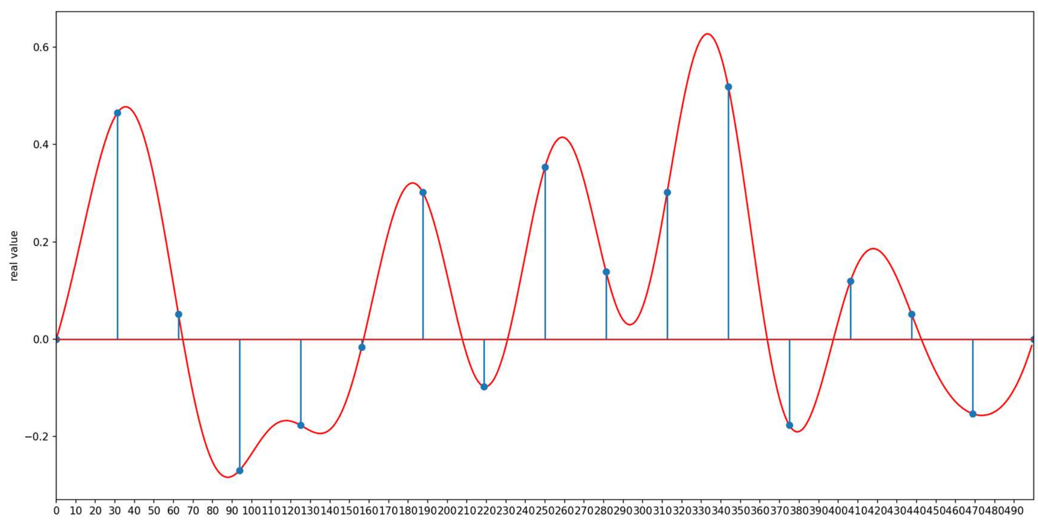
Imaginary part:



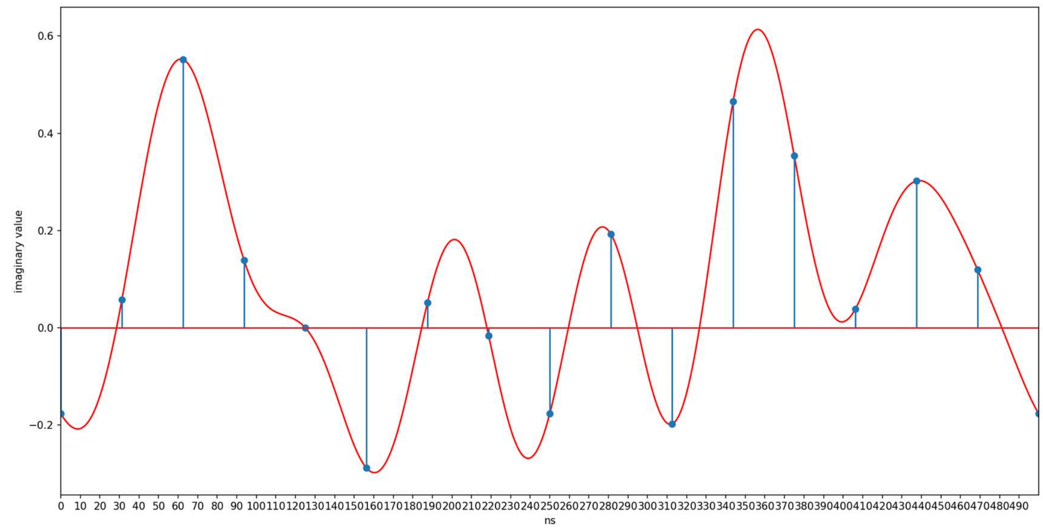
3(e)

請執行 python Q3e.py

Real part:



Imaginary part:



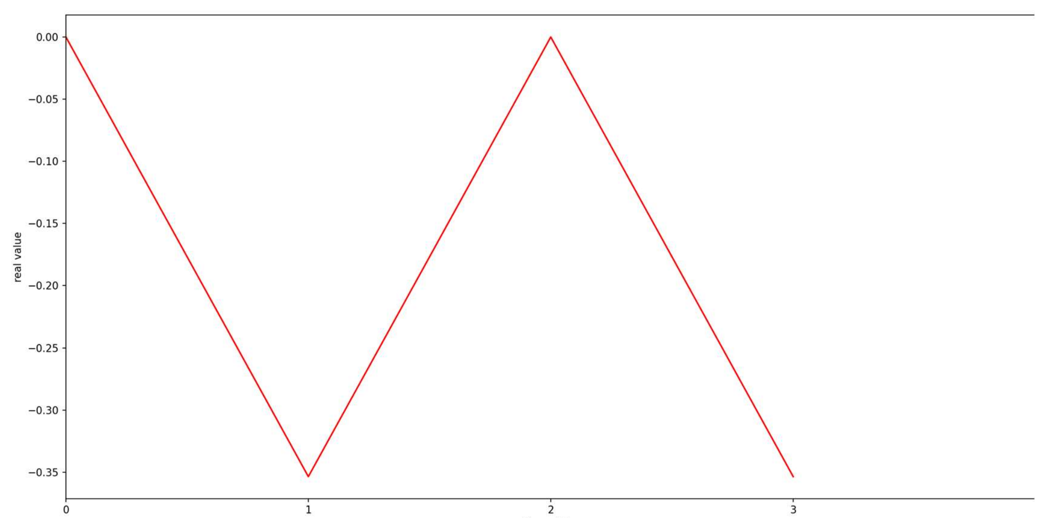
我們可以看到透過 IFFT 所呈現出來的 time-domain waveform 是藍色的離散點。而透過 3(c)所呈現的 waveform 是 continuous 的(紅色)。IFFT 所呈現的波形趨勢也是 continuous waveform 的，差別只在 IFFT 是每隔 Ts 的區間去採樣做圖。
之所以要乘以 1/8 是因為現在有 8 個 subcarrier，而離散反傅立葉轉換和採用 3(c) 的做法剛好差一個 1/N 的係數。

4

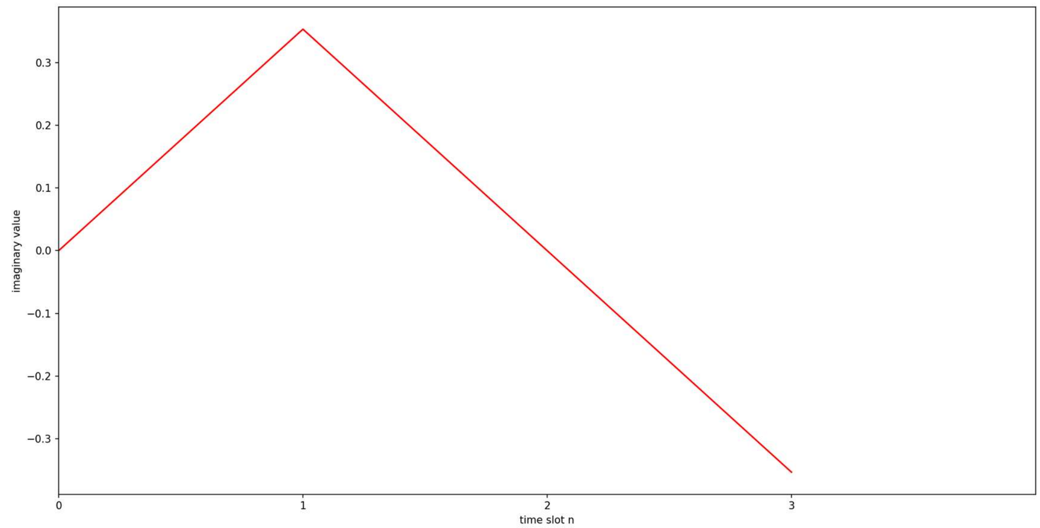
4(a)

請執行 python Q4a.py

Real part:



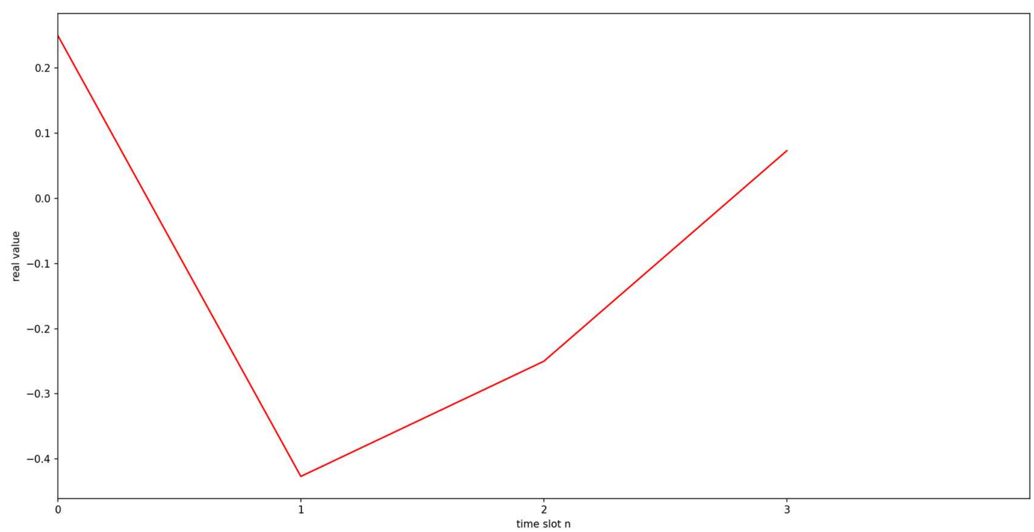
Imaginary part:



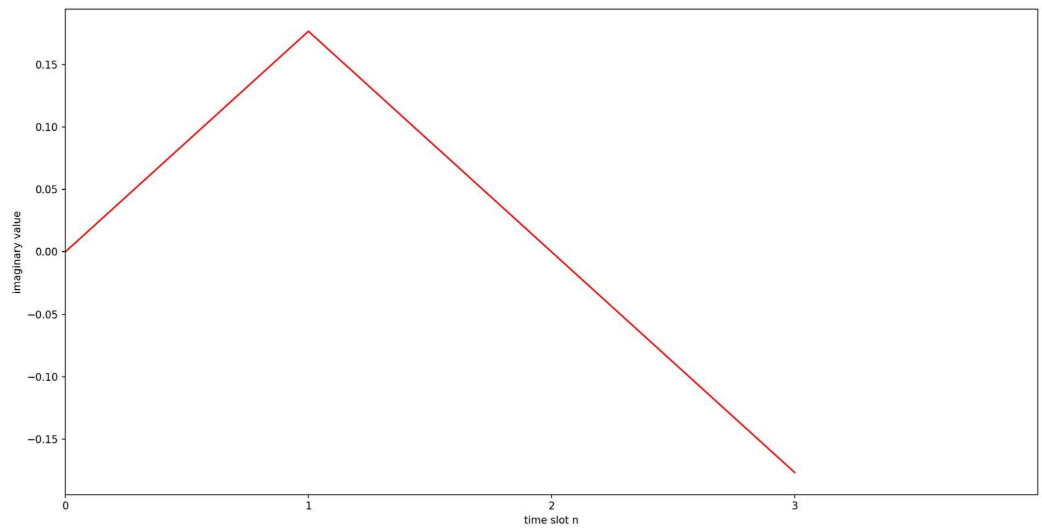
4(b)

請執行 python Q4b.py

Real part:



Imaginary part:



4(c)

我們可以觀察到 delay-Doppler domain 在有經過 ISFFT 的轉換和沒有經過 ISFFT 的轉換，最後在 time domain 輸出所呈現出不管是實部對時間的關係或是虛部對時間的關係，其波形，以及值都不一樣。

在 4(a)的做法中，delay-Doppler domain 所呈現出的是在每個 delay 的時候，其是否有 doppler 都普勒效應。因此理論上當我們經過 ISSFT 轉換到 time frequency domain 後所呈現出的是每個 time slot 時的頻譜分布。這個時候我們針對每個 timeslot 去做 IFFT 所呈現的才是那個 timeslot 的波形圖，因為 IFFT 做的正是把頻率域轉換到時間域。整個過程這樣做我們可以確保一開始把 phase 放到 delay-Doppler domain 的資訊不會隨著 channel 的傳送，最後因為都普勒效應而還原不出原本的資訊。也正是 OTFS 的做法。

在 4(b)的做法中，當我們不經過 ISSFT，直接對 delay-Doppler domain 做 IFFT，等於是我們好像把 phase 等數據放在 time frequency domain，再去做 IFFT 一樣，而不是放在 delay-Doppler domain，會變得比較像 OFDM 的架構。這樣有可能因為 channel 的傳送，產生 doppler effect，最後還原不出原本的頻譜樣貌。