

通訊系統電腦模擬與量測

Simulations and Measurements of Communication Systems

國立臺灣海洋大學 通訊與導航工程學系

Oral Report_1

系級：通訊4A

學號：0086C035

姓名：余佳駿



Experiment # 1

PN Sequence Generation and Autocorrelation



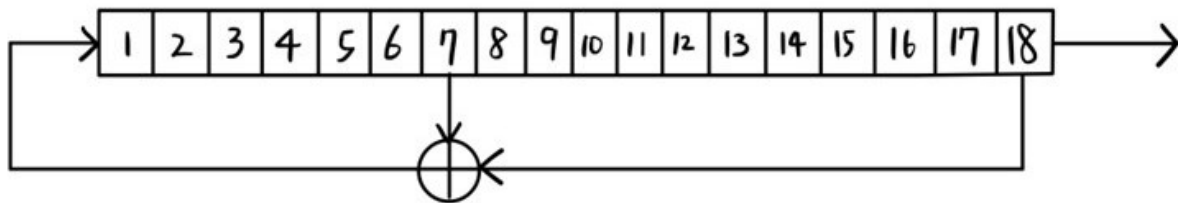
1-1 實驗介紹

- **PN序列產生**

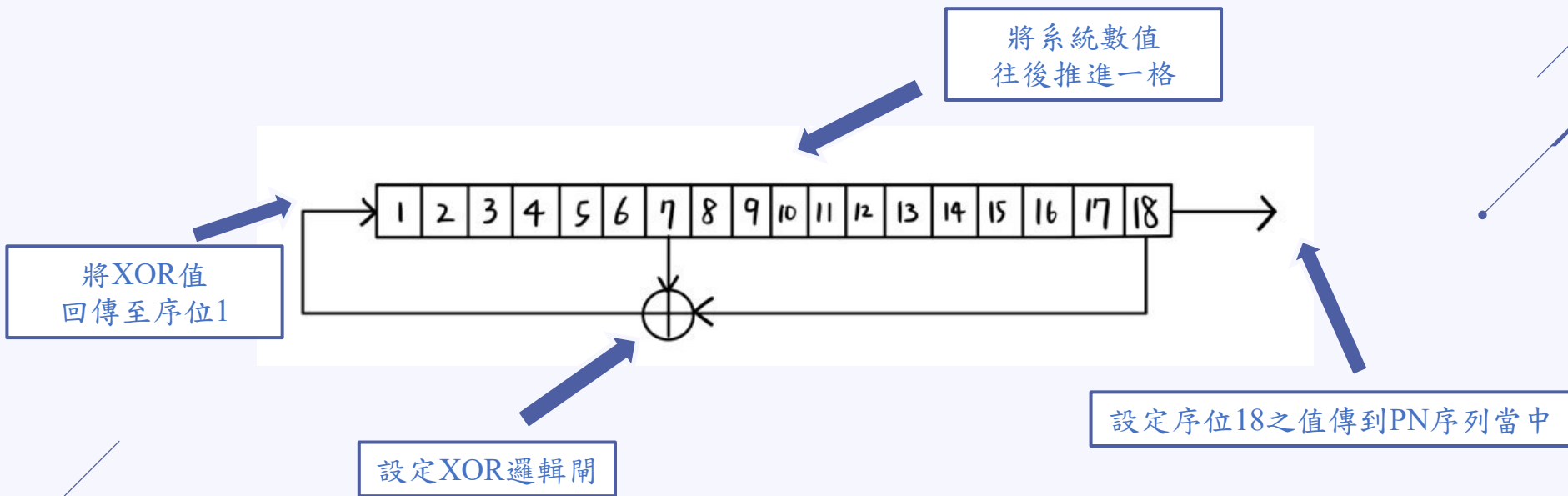
PN序列是使用初始值通過算法生成的二進制數字序列。序列具有週期性，週期長度取決於生成器中使用的移位寄存器的長度。PN序列通常使用線性反饋移位寄存器（LFSR）產生。LFSR使用一個反饋函數，將移位寄存器中的一些二進制數字組合起來生成一個新的數字，並將其移位到寄存器中。反饋函數旨在產生最大長度序列，這是一個具有給定移位寄存器長度的最大可能週期的序列。

實驗目標

1-1.設計以下LFSR系統並產生PN序列



設計想法



```
1 % Experiment_1 PN Sequence Generation
2
3 clc;clear all;
4 G=50; %設定PN序列長度
5 sd2=[1 0 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0]; %設定LFSP系統初始序列
6 PN2=[]; %創建一個PN序列容器
7
8 for i=1:G;
9     PN2=[PN2 sd2(18)];
10
11     tmp2=xor(sd2(7),sd2(18));%利用XOR邏輯閘
12
13     for j=1:17
14         k=18-j;
15         sd2(k+1)=sd2(k); %將系統往後推
16     end
17     sd2(1)=tmp2;
18 end
19 x2 = 1 - 2.*PN2; % Convert to bipolar.
20
```

Command Window

Columns 1 through 19

1	-1	1	1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1
---	----	---	---	----	---	----	----	----	----	----	----	----	----	----	----	---	----	----

Columns 20 through 38

1	-1	-1	1	1	1	1	-1	1	1	-1	-1	-1	-1	1	1	-1	1	-1
---	----	----	---	---	---	---	----	---	---	----	----	----	----	---	---	----	---	----

Columns 39 through 50

1	1	1	1	-1	1	1	1	1	-1	-1	1
---	---	---	---	----	---	---	---	---	----	----	---

fx >>

圖1.輸出結果

1-2 訊號自相關

- 訊號自相關

自相關（Autocorrelation），是一個訊號在其自身在不同時間點的互相關。在兩個時間序列之間和同一個時間序列中比較任意兩個不同時刻的相關程度，

- ZC序列特性

只要序列挪動一點，則會變得相關性就會大幅降低，因此可以用來標示不同UE，並且不容易誤判。

實驗目標

1-2.設計一個ZC sequence，並對其做自相關觀察，分別使用 Circular shift 與 Linear shift 做比較。

$$a_q(n) = \exp\left[-j\pi q \frac{n(n+1)}{N_{zc}}\right]$$

$$r_{kk}(\sigma) = \sum_{n=0}^{N_{zc}-1} a_k(n) a_k^*[(n+\sigma)] = \delta(\sigma)$$

圖2. 自相關公式

```
1 % Experiment_1-1 Autocorrelation Calculation(Linear shift)
2 clear;
3 Nzc=997;
4 q=Nzc/31;%32.16
5 q1=floor(q+0.5); %32
6
7 %生成ZC序列
8 for m = 1:Nzc
9     zc1(m)=exp(-j*pi*q1*m*(m+1)/Nzc); % 4G PSS
10 end
11
12 %做位移自相關，測試linear shift
13 for m = 1:Nzc
14
15     zc1_zero=[zeros(1,m-1) zc1(1:end-m+1)]; %位移補零
16
17     cor_zc1(m)=sum(zc1.*conj(zc1_zero))/Nzc;
18 end
19
20 t=1:997;
21 plot(t,abs(cor_zc1),'b'); % abs() %求絕對值
```

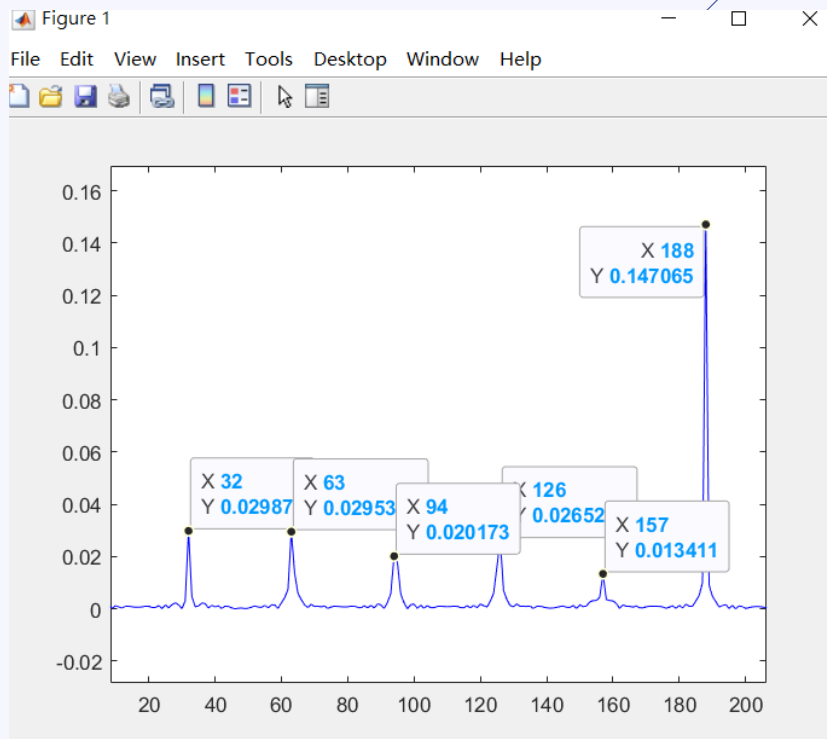
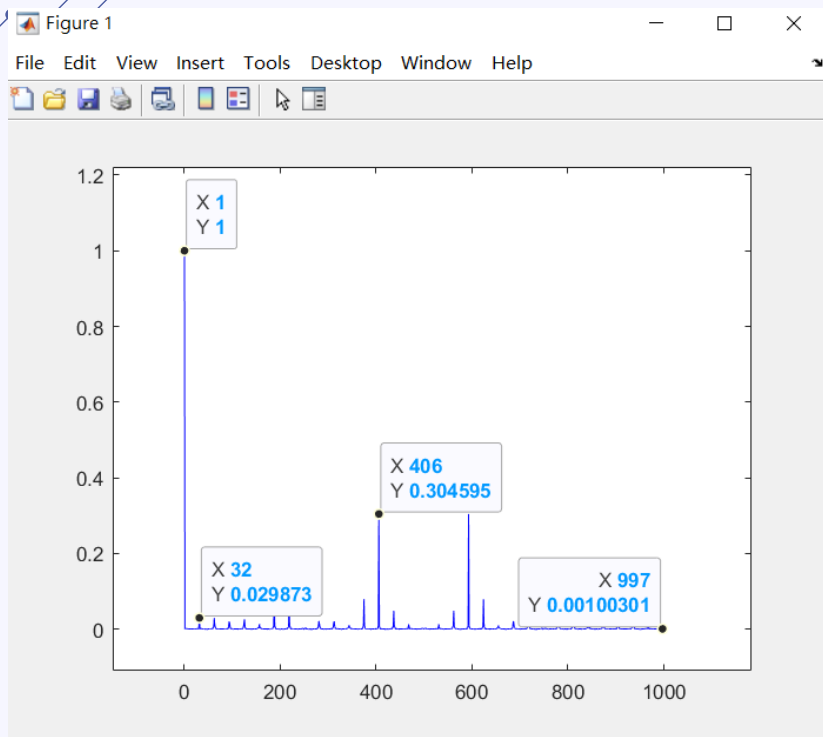


圖3. Linear shift自相關輸出結果

```

27 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
28 % Experiment_1-2 Autocorrelation Calculation (circular shift)
29 clear;
30 Nzc=997;
31 q=Nzc/31;%32.16
32 q1=floor(q+0.5); %32
33
34 %生成ZC序列
35 for m = 1:Nzc
36     zc1(m)=exp(-j*pi*q1*m*(m+1)/Nzc);%4G PSS
37 end
38
39 %做初始自相關
40 cor_zc1(1)=sum(zc1.*conj(zc1))/Nzc; %conj(zc1):返回zc1的複共軛; [.*]:按元素乘法; sum():數組元素求和
41
42 %做位移自相關, circular shift
43 for m = 2:Nzc
44     zc1_round=[zc1(m:end) zc1(1:m-1)];
45     cor_zc1(m)=sum(zc1.*conj(zc1_round))/Nzc;
46 end
47
48 t=1:997;
49 plot(t,abs(cor_zc1),'b'); % abs() %求絕對值

```

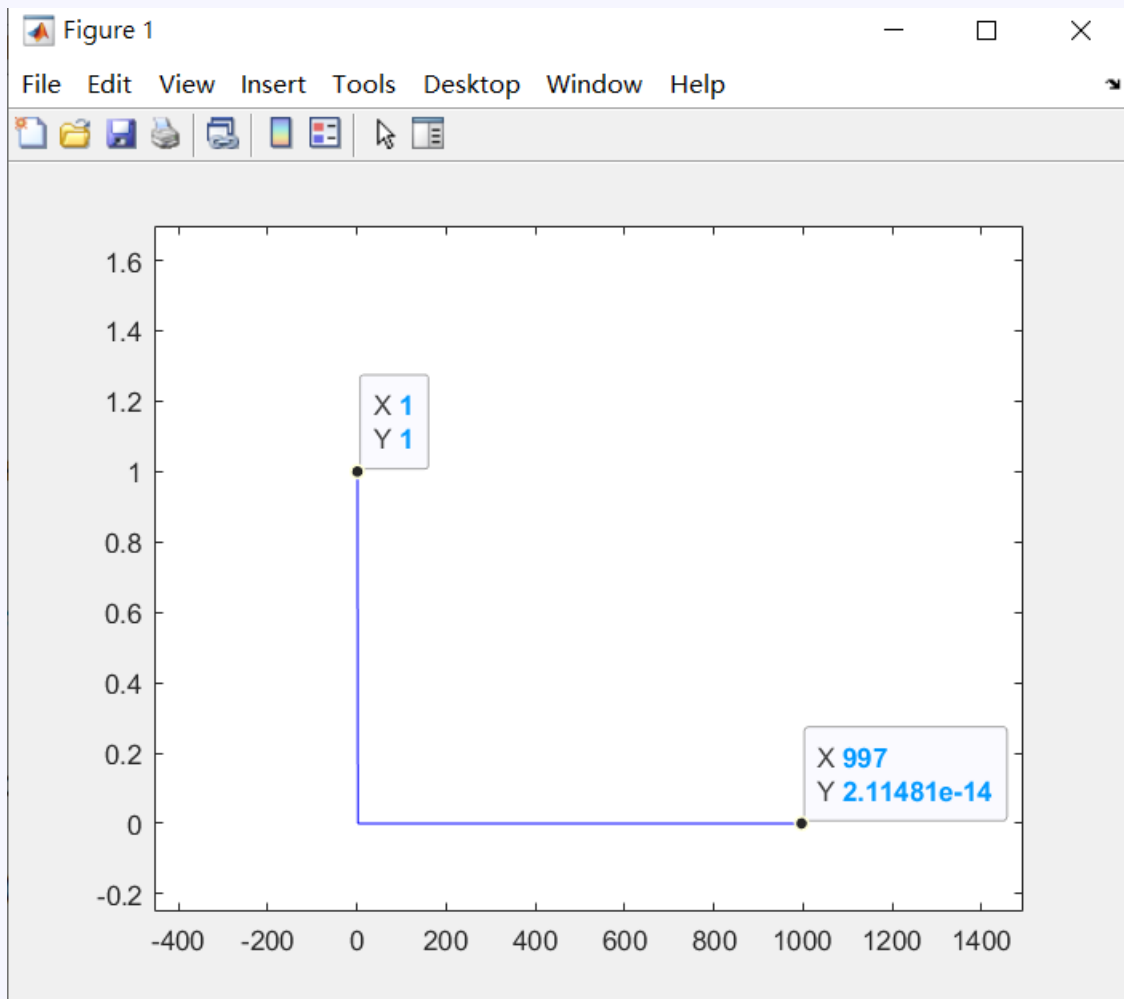
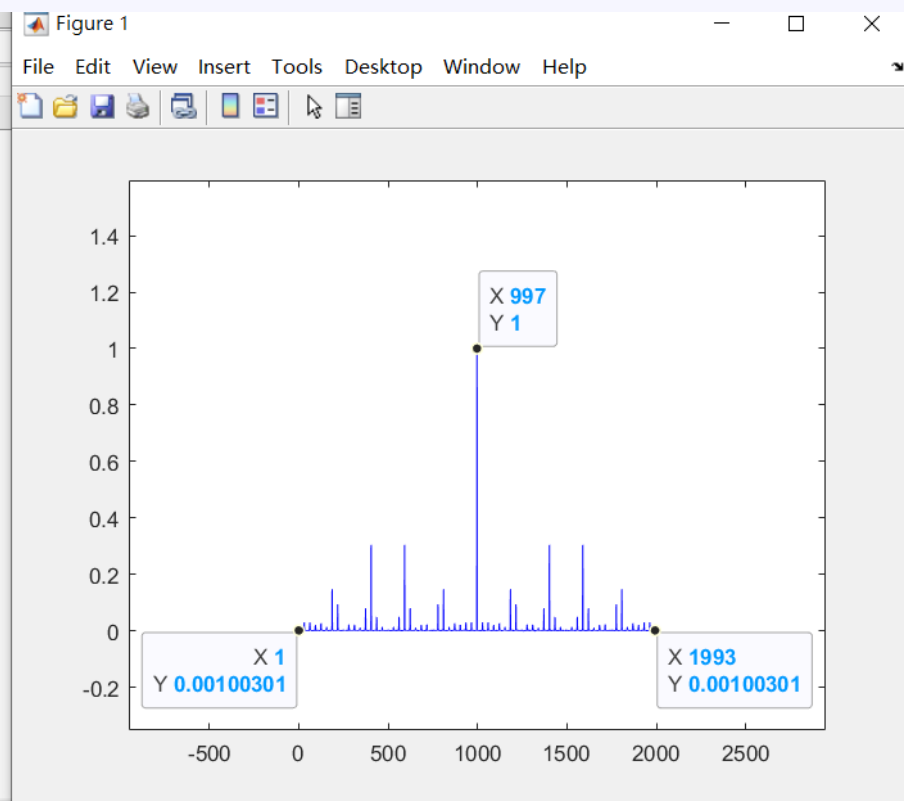


圖3. Circular shift自相關輸出結果

使用內建`xcorr()`函數觀察，會發現數值會在 $N_{zc}=997$ 時最大，這是因為使用此函數，經不斷位移，在 $N_{zc}=997$ 時會完全重疊時相關性最強

```
NAVIGATE CODE ANALYZE SECTION
> jj200 > Desktop > chun
Editor - C:\Users\jj200\Desktop\chun\test.m
PN.m x Autocorrelation.m x test.m x untitled2.m x +
1 % Experiment_1-1 Autocorrelation Calculation
2 clear;
3 Nzc=997;
4 q=Nzc/31;%32.16
5 q1=floor(q+0.5); %32
6
7 %生成ZC序列
8 for m = 1:Nzc
9     zc1(m)=exp(-j*pi*q1*m*(m+1)/Nzc); % 4G PSS
10 end
11
12 cor_zc1=xcorr(zc1)/997;
13
14 % 做位移自相關，測試linear shift
15 % for m = 1:Nzc
16 %
17 %     zc1_zero=[zeros(1,m-1) zc1(1:end-m+1)]; %位移補零
18 %
19 %     cor_zc1(m)=sum(zc1.*conj(zc1_zero))/Nzc;
20 % end
```



1-3 OVSF Generation

- OVSF是Orthogonal Variable Spreading Factor的縮寫，稱為正交可變展頻因子，系統根據展頻因子的大小給用戶分配資源，數值越大，提供的頻寬越小。

實驗目標

1-3.設計一個OVSF sequence，規則如下圖所示

3G OVSF Code

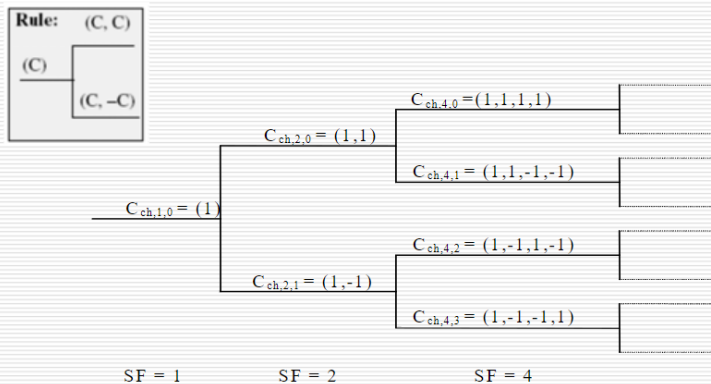


圖4. OVSF架構


```

1 function Code = OVFS(SF)
2   x="Input the SF value : ";
3   SF=input(x);
4
5   Code=1; %初始化傳的數值
6   Cnew=[]; %初始化一個暫存容器
7
8   for n=1:log2(SF) %SF=1,2,8,16...
9       Nrow=2^n; %設定分支數量
10
11       if Nrow == 2 %如果只有兩個分支
12           Cnew = [Code Code ;Code -Code]; %OVFS規則[1 1 ; 1 -1]
13           Code = Cnew; %儲存
14           Cnew = []; %清空暫存容器
15       else
16           for ind=1:Nrow %Nrow=4,8,16...
17               Cnew(ind,:)=[Code(ceil(ind/2),:),(-1)^(ind+1)*Code(ceil(ind/2),:)];
18           end
19           Code = Cnew; %儲存
20           Cnew = []; %清空暫存容器
21       end
22   end
23   Code = mat2cell(Code,ones(1,SF),SF); %將Code拆成分支 %C = mat2cell(A,dim1Dist,...,dimNDist) A
24   %Code 轉換為 1xSF 的數組，共有SF個
25   end

```

$C = \text{mat2cell}(A, [10 \ 20 \ 30], [25 \ 25])$

10x25	10x25
20x25	20x25
30x25	30x25

```
>> OVSF
```

```
fx Input the SF value : 8
```

Command Window

```
ans =
```

```
8×1 cell array
```

```
{[ 1 1 1 1 1 1 1 1]}
{[1 1 1 1 -1 -1 -1 -1]}
{[1 1 -1 -1 1 1 -1 -1]}
{[1 1 -1 -1 -1 -1 1 1]}
{[1 -1 1 -1 1 -1 1 -1]}
{[1 -1 1 -1 -1 1 -1 1]}
{[1 -1 -1 1 1 -1 -1 1]}
{[1 -1 -1 1 -1 1 1 -1]}
```

```
fx >>
```

3G OVSF Code

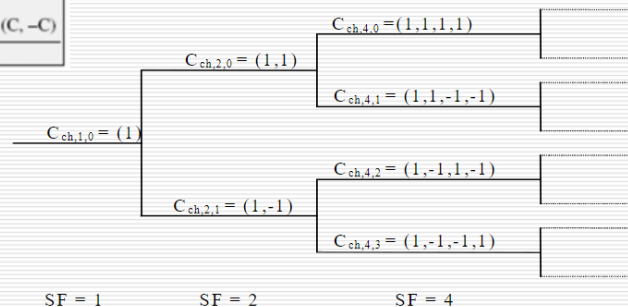
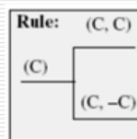


圖5. OVSF輸出結果



Experiment #2



Linear Convolution and Circular Convolution

2-1. Test the length of Linear Convolution.

2-2. Test the equivalent condition of Linear Convolution with the Circular Convolution.

```
Editor - C:\Users\jj200\Desktop\chun\LC.m
PN.m x Autocorrelation.m x OVFS.m x dftmtx.m x test.m x QPSK.m x untitled * x LC.m x +
1 % 驗證線性卷積的長度
2 % 驗證循環卷積與線性卷積之等效條件，在進行DFT 之前，必須用零填充向量
3 % 使長度至少為  $N + L - 1$ 。對DFT 的積求逆轉換後，只保留前  $N + L - 1$  個元素。
4
5 clear;
6
7 x = [5 6 8 2 5]; %L=5
8 y = [6 -1 3 5 1]; %M=5
9
10 %Linear Convolution
11 clin = conv(x,y,'full');
12 str=['The length of Linear Convolution is : ',num2str(length(clin))];% 驗證線性卷積的長度
13 disp(str); % length=L+M-1
14
15 %Circular Convolution
16 %由於輸出長度為  $L+M-1=9$ ，需用 0 填充 x,y 兩個向量，把長度都補滿9
17 xpad = [x zeros(1,9-length(x))];
18 ypad = [y zeros(1,9-length(y))];
19 ccirc = ifft(fft(xpad).*fft(ypad)); % Circular Convolution公式
20
21 %繪製線性卷積
22 subplot(2,1,1) %2列1行之區域1
23 stem(clin, 'filled' )
24 ylim([0 100]) % ylim:設置y 坐標軸範圍
25 title( 'Linear Convolution of x and y' )
26
27 %繪製循環卷積
28 subplot(2,1,2) %2列1行之區域2
29 stem(ccirc, 'filled' )
30 ylim([0 100])
31 title( 'Circular Convolution of xpad and ypad' )
32
33 %填零後的向量 xpad 和 ypad 的循環卷積等效於 x 和 y 的線性卷積。
34 %因為輸出長度皆為  $4+3-1$ ，並保留了循環卷積所有元素。
```

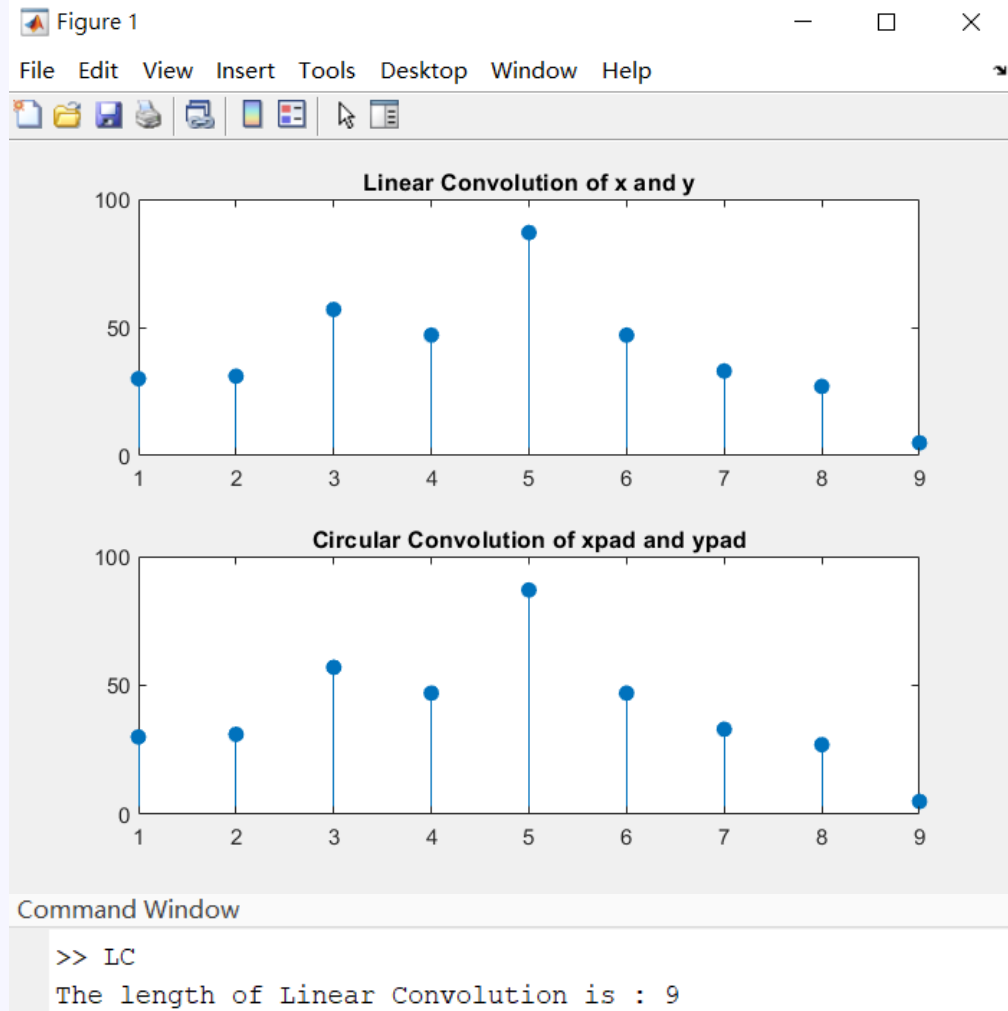
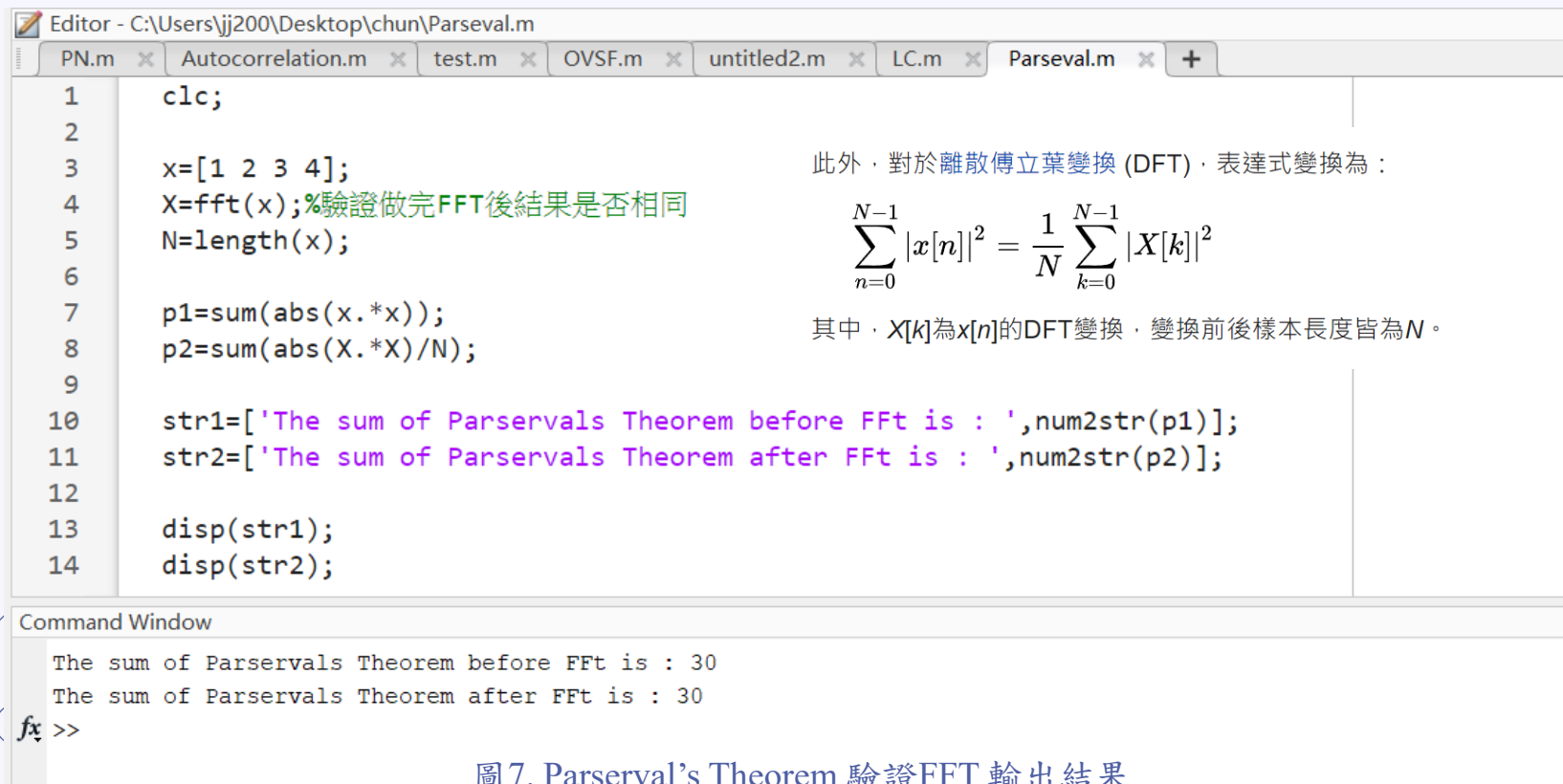


圖6. Linear Convolution 長度輸出結果

2-3. Test the Parseval's Theorem



Editor - C:\Users\jj200\Desktop\chun\Parseval.m

PN.m x Autocorrelation.m x test.m x OVSF.m x untitled2.m x LC.m x Parseval.m x +

```
1  clc;
2
3  x=[1 2 3 4];
4  X=fft(x);%驗證做完FFT後結果是否相同
5  N=length(x);
6
7  p1=sum(abs(x.*x));
8  p2=sum(abs(X.*X)/N);
9
10 str1=['The sum of Parsevals Theorem before FFt is : ',num2str(p1)];
11 str2=['The sum of Parsevals Theorem after FFt is : ',num2str(p2)];
12
13 disp(str1);
14 disp(str2);
```

Command Window

```
The sum of Parsevals Theorem before FFt is : 30
The sum of Parsevals Theorem after FFt is : 30
fx >>
```

此外，對於離散傅立葉變換 (DFT)，表達式變換為：

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2$$

其中， $X[k]$ 為 $x[n]$ 的DFT變換，變換前後樣本長度皆為 N 。

圖7. Parseval's Theorem 驗證FFT 輸出結果

2-4. Test the properties of the DFT matrix

□ The DFT operator can be expressed in matrix form

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N & W_N^2 & \dots & W_N^{(N-1)} \\ 1 & W_N^2 & W_N^4 & \dots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{(N-1)} & W_N^{(N-1)2} & \dots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

$$\omega_N = e^{-2\pi i/N}$$

圖8. DFT matrix 數學表示式

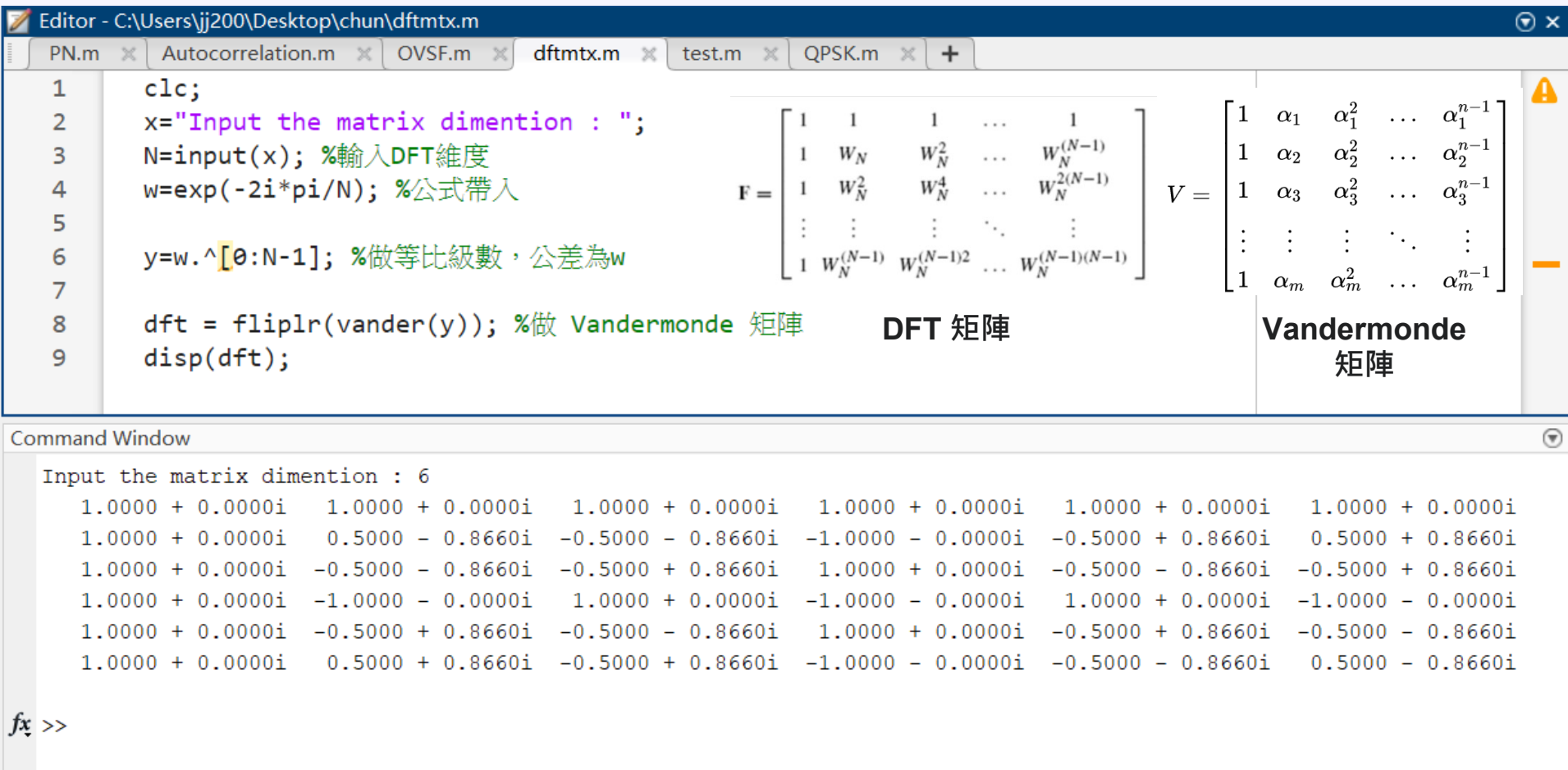


圖9. DFT matrix輸出結果

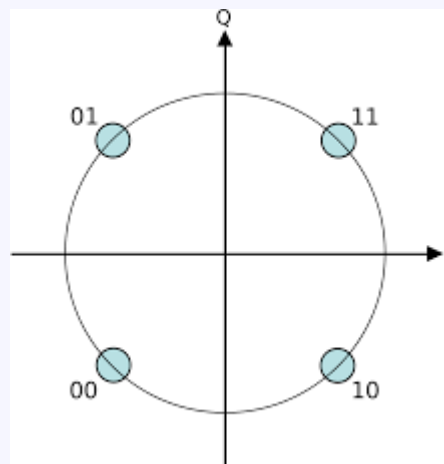
Experiment #3 #4

BER Performance of
Antipodal Binary Sequence

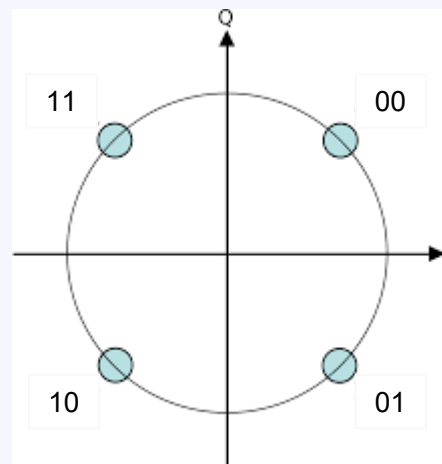
BER Performance
with Frequency Offset

3-1.設計QPSK系統，並與加入AWGN之訊號進行比較

設計想法，產生一組0、1隨機高斯分布序列，序列當中倆倆一組分別為實部與虛部，在傳送端利用循環條件式映射成星座圖，之後再加入AWGN與頻率偏移進行比較，並在接收端利用循環條件式解回序列，並分析無雜訊、加入雜訊、以及頻偏之訊號BER，並繪製成星座圖，之後也近一步對於星座圖是否使用Gray Code進行比較。



Gray Code



Non Gray Code

```
Editor - C:\Users\jj200\Desktop\ComSIM\QPSK.m
untitled.m x ASK.m x BPSK.m x FSK.m x test.m x QPSK.m x QPSK_offset.m x tttt.r

1 clear all;
2 clc;
3
4 SNR_dB=1:20; %設定訊雜比範圍，此為dB
5 snr=10.^(SNR_dB/10); %將訊雜比從dB轉化linear去算SNR
6 N=100000; %取樣次數
7 df_T=0.1;
8 A=randi([0,1],1,N); %創建一個從0跟1隨機取N次的序列A
9 [row,column] = size(A); %將A的行列數值分別存放 row=1,column=100000
10
11 y=[]; %創建一個容器，等等放QPSK
12 for it=1:row %將序列轉換後映射至星座圖
13     p=1;
14     for t = 1:2:column % 1到100000，一次讀2個，共有50000組
15         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16         % TX symbols
17         % matlab ind 從1開始
18         % QPSK，使用Gray Code
19         if A(1,t)==0 && A(1,t+1)==0 %若A(1,1)=0 且 A(1,2)=0
20             y(1,p)=-1-1i;
21
22         elseif A(1,t)==0 && A(1,t+1)==1 %若A(1,1)=0 且 A(1,2)=1
23             y(1,p)=-1+1i;
24
25         elseif A(1,t)==1 && A(1,t+1)==0 %若A(1,1)=1 且 A(1,2)=0
26             y(1,p)=1-1i;
27
28         else
29             y(1,p)=1+1i; %若A(1,1)=1 且 A(1,2)=1
30
31     end
20
```



```

73 % RX symbols
74 % QPSK轉回序列 (無頻差)
75 [n,m] = size(ys); %將ys的行列數值分別存放 n=1,m=50000(2個bits一組)
76 q1=[]; %創建一個容器，等等解回之序列
77 q2=[];
78 for b=1:n % 1:1 % 利用循環條件式解回序列 (將4個星座點位置輸出為01序列)
79     j=1;
80     %利用星座點象限位置判定01序列
81     for d=1:m % 1:50000
82         if real(ys(b,d))<0 && imag(ys(b,d))<0 % real():取實部, imag():取虛部
83             q1(b,j)=0;
84             q1(b,j+1)=0;
85
86         elseif real(ys(b,d))<0 && imag(ys(b,d))>0
87             q1(b,j)=0;
88             q1(b,j+1)=1;
89
90         elseif real(ys(b,d))>0 && imag(ys(b,d))<0
91             q1(b,j)=1;
92             q1(b,j+1)=0;
93
94         elseif real(ys(b,d))>0 && imag(ys(b,d))>0
95             q1(b,j)=1;
96             q1(b,j+1)=1;
97         end
98 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
99 % RX symbols
100 % QPSK轉回序列 (有頻差)
101     if real(ys_off(b,d))<0 && imag(ys_off(b,d))<0
102         q2(b,j)=0;
103         q2(b,j+1)=0;
104
105     elseif real(ys_off(b,d))<0 && imag(ys_off(b,d))>0
106         q2(b,j)=0;
107         q2(b,j+1)=1;
108
109     elseif real(ys_off(b,d))>0 && imag(ys_off(b,d))<0
110         q2(b,j)=1;
111         q2(b,j+1)=0;
112
113     elseif real(ys_off(b,d))>0 && imag(ys_off(b,d))>0
114         q2(b,j)=1;
115         q2(b,j+1)=1;
116
117     end
118     j=j+2; %2個一組進行判定，所以完成一次判定後往後推2格(新的一組)
119 end
120 end

```

```

122 [number1,BER_AWGN(count)] = symerr(A,q1); % BER_AWGN:加入AWGN之QPSK之BER
123 % 計算BER_AWGN，公式如下
124 % symerr(X,Y) = abs(X-Y)/abs(Y)
125
126 [number2,BER_AWGN_off(count)] = symerr(A,q2); %BER_AWGN_off
127
128 [number3,BER_Initial(count)] = symerr(A,A); %BER_Initial
129
130 end
131
132 % BER_Initial=1/2*erfc(sqrt(snr/2)); %計算理論QPSK之BER
133
134 %計算平均位元錯誤率
135 BER_avg_I=sum(BER_Initial)/count;
136 BER_avg_N=sum(BER_AWGN)/count;
137 BER_avg_O=sum(BER_AWGN_off)/count;
138
139 %繪製理論之QPSK
140 scatterplot(y);
141 title({'QPSK Initial','Average BER : ',num2str(BER_avg_I)});
142 grid on;
143 axis tight;
144
145 %繪製加入AWGN之QPSK
146 scatterplot(ys);
147 title({'QPSK AWGN','Average BER : ',num2str(BER_avg_N)});
148 grid on;
149 axis tight;
150
151 %繪製頻偏且加入AWGN之QPSK
152 scatterplot(ys_off);
153 title({'QPSK AWGN Offset','Average BER : ',num2str(BER_avg_O)});
154 grid on;
155 axis tight;
156
157 %繪圖
158 figure;
159 % plot(SNR_dB,BER_Initial,'-g','linewidth',2);hold on;
160 semilogy(SNR_dB,BER_AWGN,'-b','linewidth',2);hold on;
161 semilogy(SNR_dB,BER_AWGN_off,'-r','linewidth',2);hold on;
162 legend('QPSK+AWGN','頻偏QPSK+AWGN');
163 axis([-1,10,10^-4,1]);
164 title('BER Performance of the QPSK');
165 xlabel('SNR(dB)');
166 ylabel('BER');

```

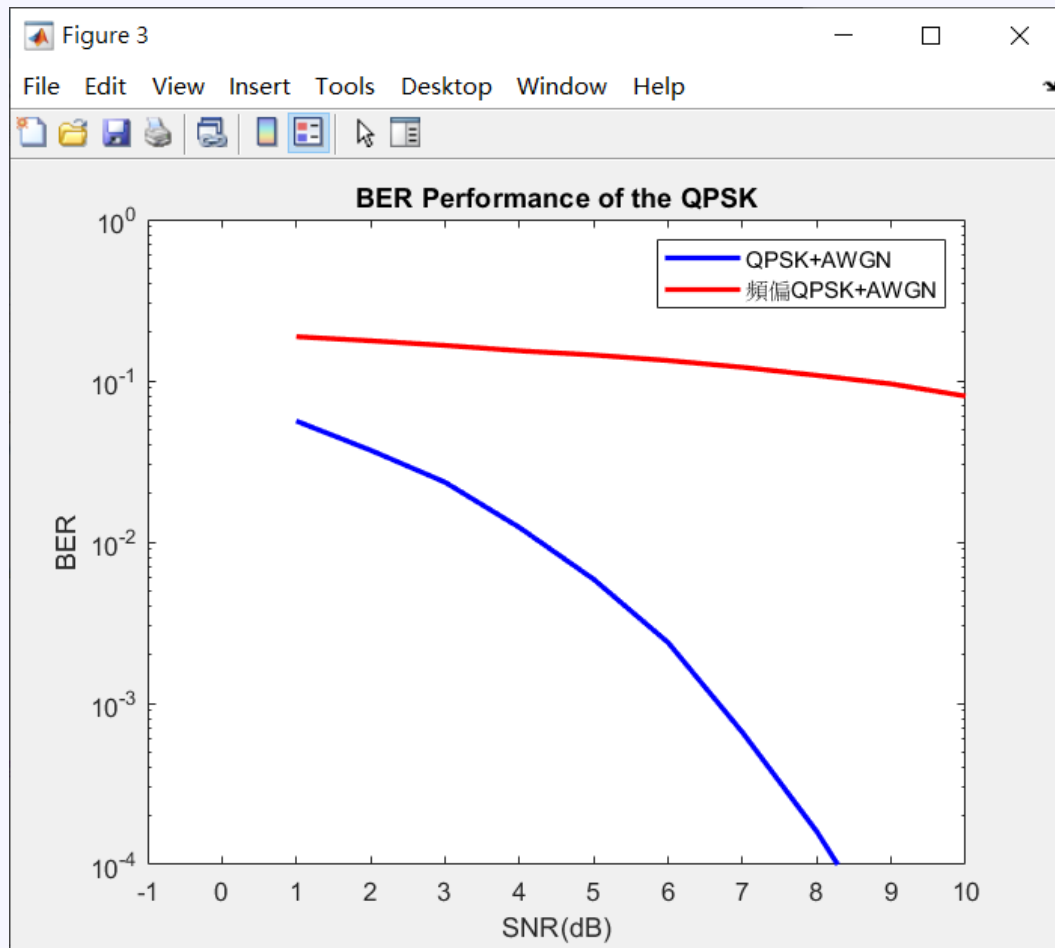


圖10. QPSK BER 比較圖(使用Gray Code)

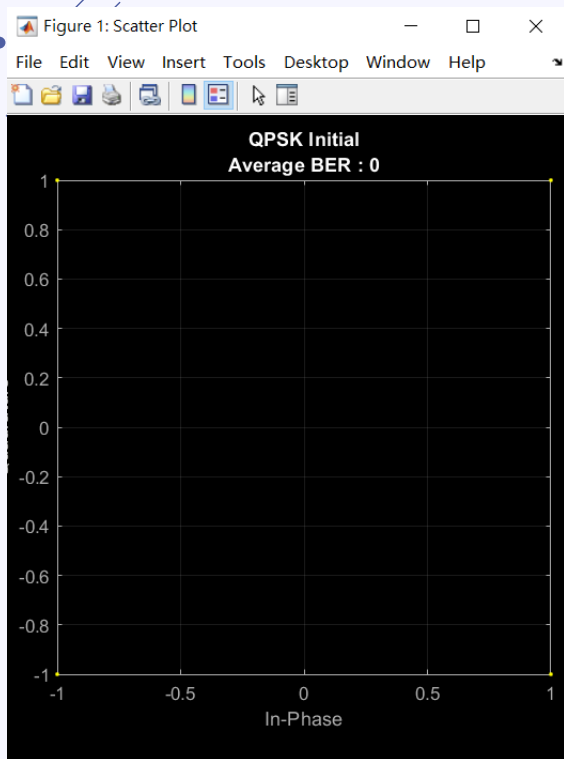


圖11. QPSK 星座圖
(無雜訊)

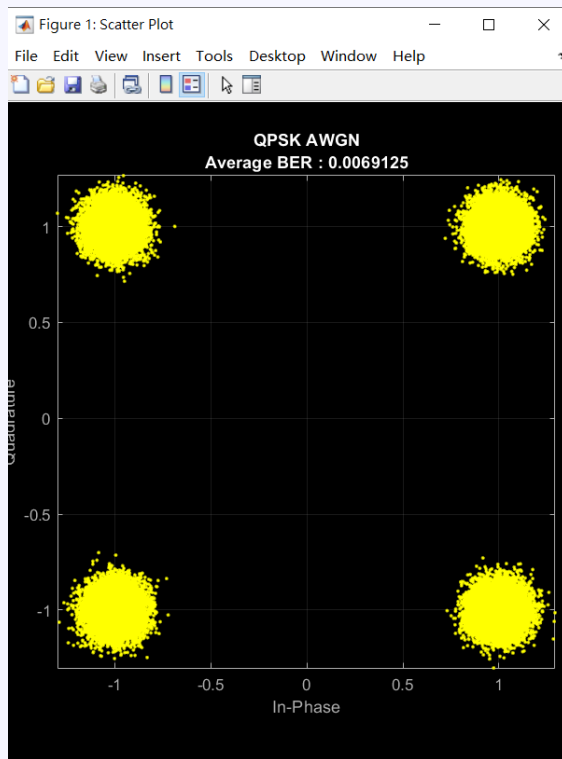


圖12. QPSK 星座圖
(加入AWGN雜訊)

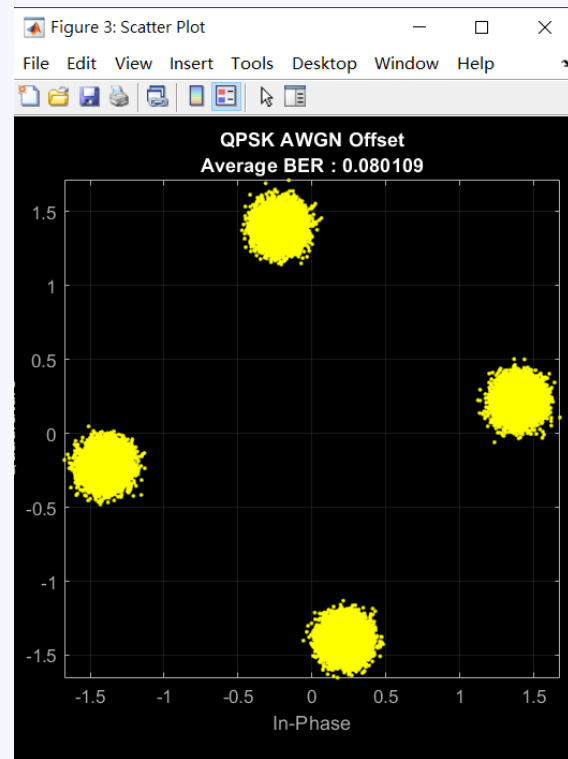


圖13. QPSK 星座圖
(頻偏後加入AWGN雜訊)


```
% TX symbols
% matlab ind 從1開始
% QPSK, 不使用Gray Code
if A(1,t)==0 && A(1,t+1)==0      %若A(1,1)=0 且 A(1,2)=0
    y(1,p)=1+1i;

elseif A(1,t)==0 && A(1,t+1)==1  %若A(1,1)=0 且 A(1,2)=1
    y(1,p)=1-1i;

elseif A(1,t)==1 && A(1,t+1)==0  %若A(1,1)=1 且 A(1,2)=0
    y(1,p)=-1-1i;

else
    y(1,p)=-1+1i;                %若A(1,1)=1 且 A(1,2)=1

end

p=p+1;
end
```

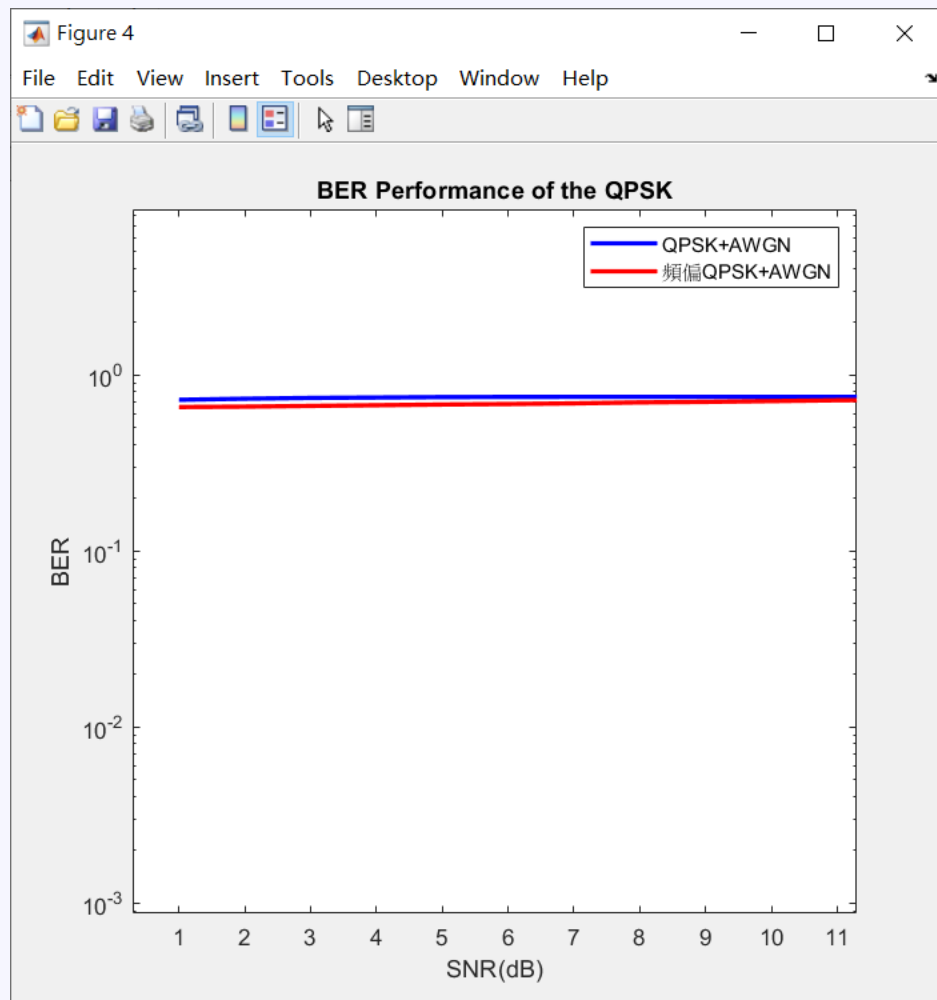


圖14. QPSK BER 比較圖(不使用Gray Code)

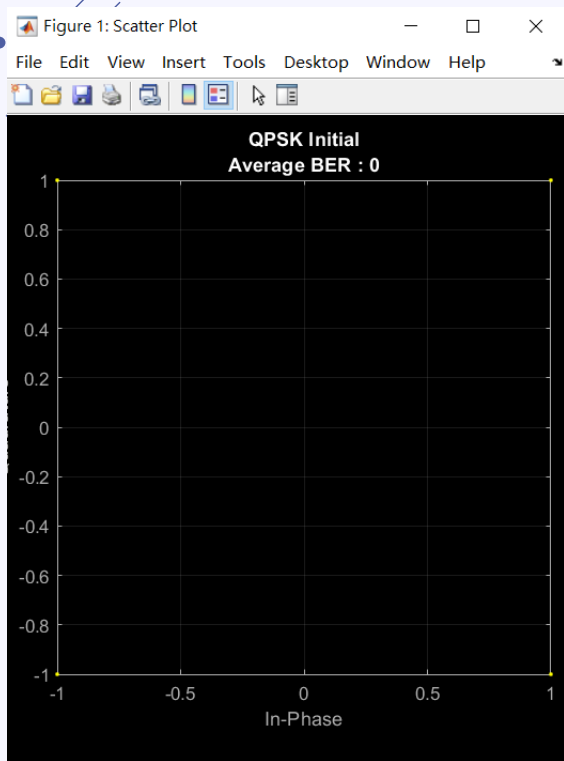


圖15. QPSK 星座圖
(無雜訊)

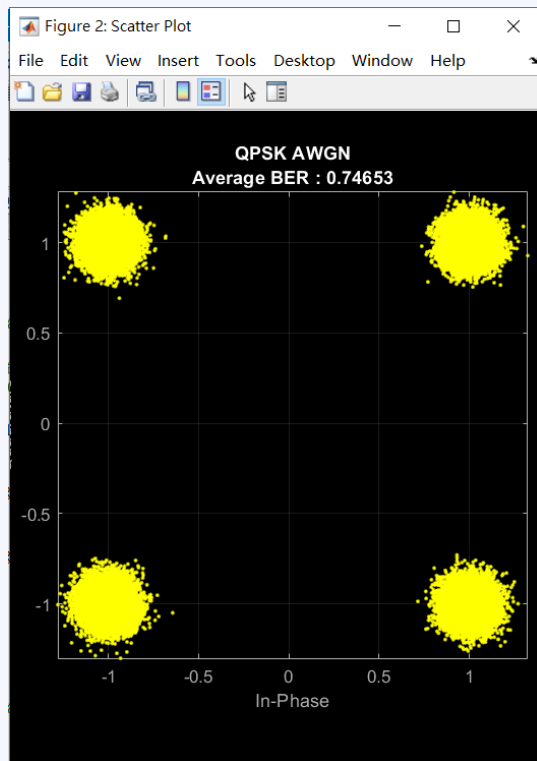


圖16. QPSK 星座圖
(加入AWGN雜訊)

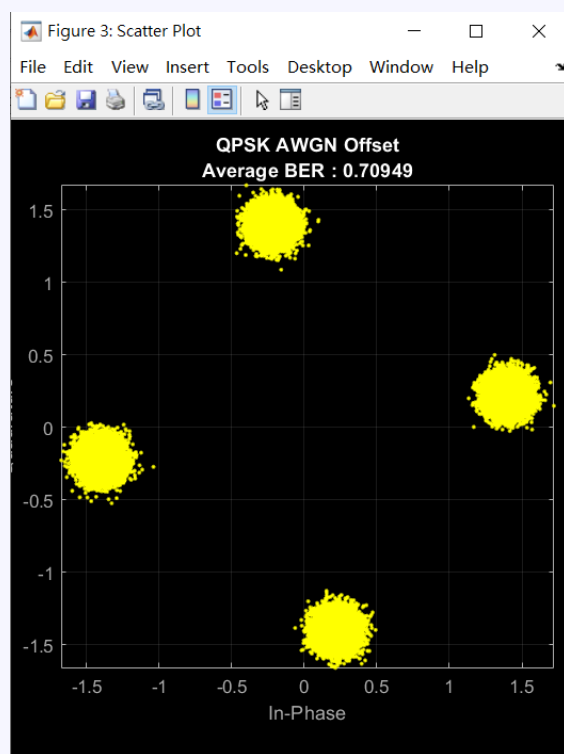


圖17. QPSK 星座圖
(頻偏後加入AWGN雜訊)

3-2.設計BPSK系統

```
Editor - C:\Users\jj200\Desktop\ComSIM\BPSK.m
untitled.m x BPSK.m x ASK.m x FSK.m x test.m x tttt.m x +
1  clc
2  clear all;
3
4  f = 1;      %sin函數頻率
5  fs = 100;   %取樣頻率
6  t = 0:1/fs:1; %時間間隔
7  snr=15;     %訊雜比
8  df_T=0.3;
9
10 %設定BPSK相位差
11 p1 = 0;
12 p2 = pi;
13
14 %設定位元數
15 N = 10;
16
17 %生成隨機訊號
18 bit_sequence=round(rand(1,N));
19
20 %創建容器
21 time = [];
22 digital_signal = [];
23 PSK = [];
24 carrier_signal = [];
```

```
26 %產生位元序列
27 for ii = 1:1:N %1:10
28
29     if bit_sequence(ii) == 0
30         bit = zeros(1,length(t));
31     else
32         bit = ones(1,length(t));
33     end
34     digital_signal = [digital_signal bit];
35
36     %產生BPSK訊號
37     if bit_sequence(ii) == 0
38         bit = sin(2*pi*f*t+p1);
39     else
40         bit = sin(2*pi*f*t+p2);
41     end
42     PSK = [PSK bit];
43
44     %生成載波
45     carrier = sin(2*pi*f*t); % f=1 % t=0:1/fs:1
46     carrier_signal = [carrier_signal carrier];
47     time = [time t]; %時間間隔 1/fs =0.01
48     t = t + 1;
49 end
```

```
51 %加入AWGN Noise
52 PSK_AWGN=awgn(PSK,snr);
53
54 %Channel fading coefficients
55 K=exp(-j*2*pi*df_T);
56 PSK_off=PSK*K;
57
58 %頻偏後加入AWGN Noise
59 PSK_offset=awgn(PSK_off,snr);
60
61 %計算理想BER
62 BER_BPSK=1/2*erfc(sqrt(snr)); %補誤差函數
63
64 % 計算加入雜訊之BER
65 BER_BPSK_AWGN=abs(PSK_AWGN-PSK)/abs(PSK);
66
67 % 計算頻偏後加入雜訊之BER
68 BER_BPSK_offset=abs(PSK_offset-PSK)/abs(PSK);
```

$$Q(z) = \frac{1}{2} \operatorname{erfc}\left(\frac{z}{\sqrt{2}}\right)$$

```
71 %繪圖
72 subplot(5,1,1);
73 plot(time,PSK,'r','linewidth',2);
74 title(['理想 BPSK 調變訊號','BER=',num2str(BER_BPSK)])
75 grid on;
76 axis tight;
77
78 subplot(5,1,2);
79 plot(time,PSK_AWGN,'r','linewidth',2);
80 title(['BPSK+AWGN 調變訊號','BER=',num2str(BER_BPSK_AWGN)])
81 grid on;
82 axis tight;
83
84 subplot(5,1,3);
85 plot(time,PSK_offset,'r','linewidth',2);
86 title(['頻偏之 BPSK+AWGN 調變訊號','BER=',num2str(BER_BPSK_offset)])
87 grid on;
88 axis tight;
89
90 subplot(5,1,4);
91 plot(time,digital_signal,'linewidth',2);% linewidth:調整線條粗度
92 title('位元序列')
93 grid on;
94 axis([0 time(end) -0.5 1.5]);
95
96 subplot(5,1,5);
97 plot(time,carrier_signal,'linewidth',2);
98 title('載波訊號')
99 grid on;
100 axis tight;
```

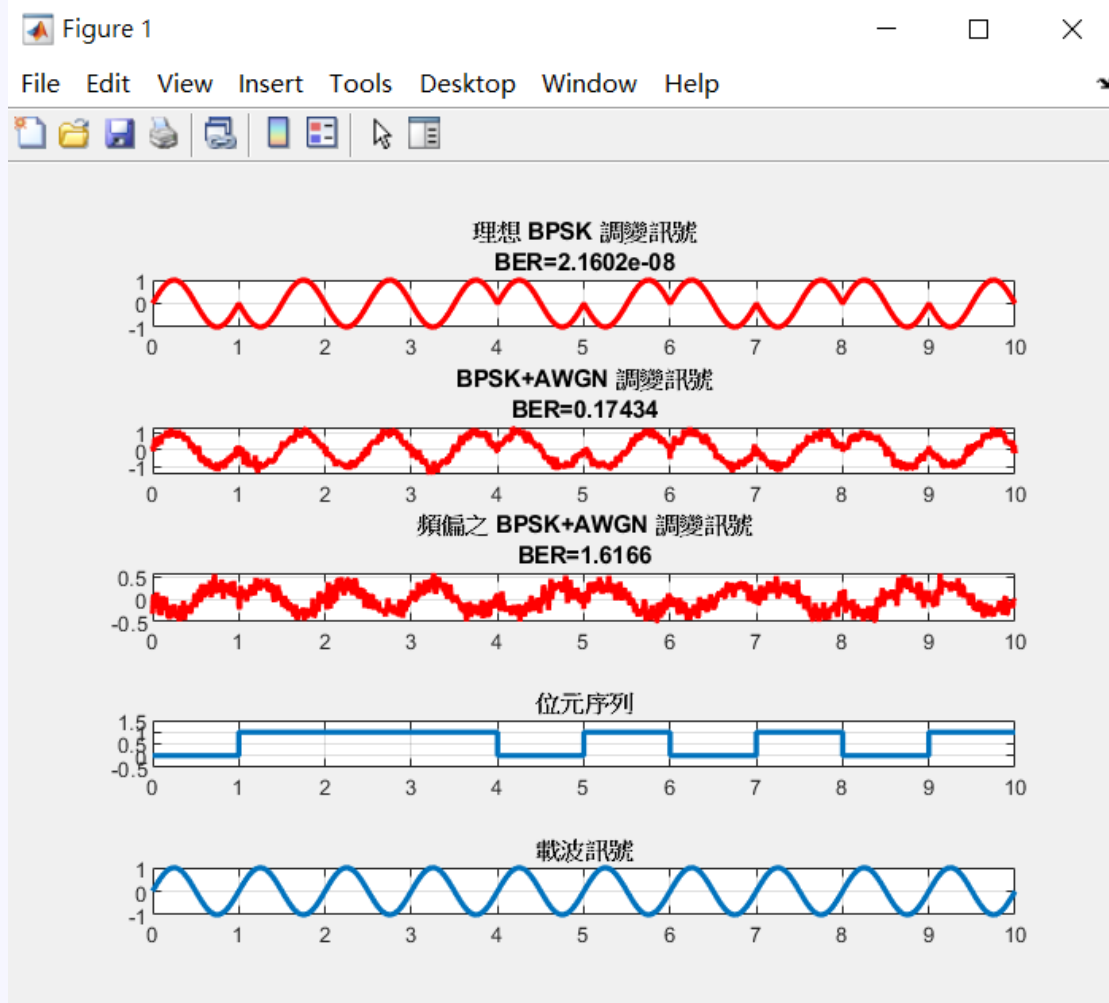
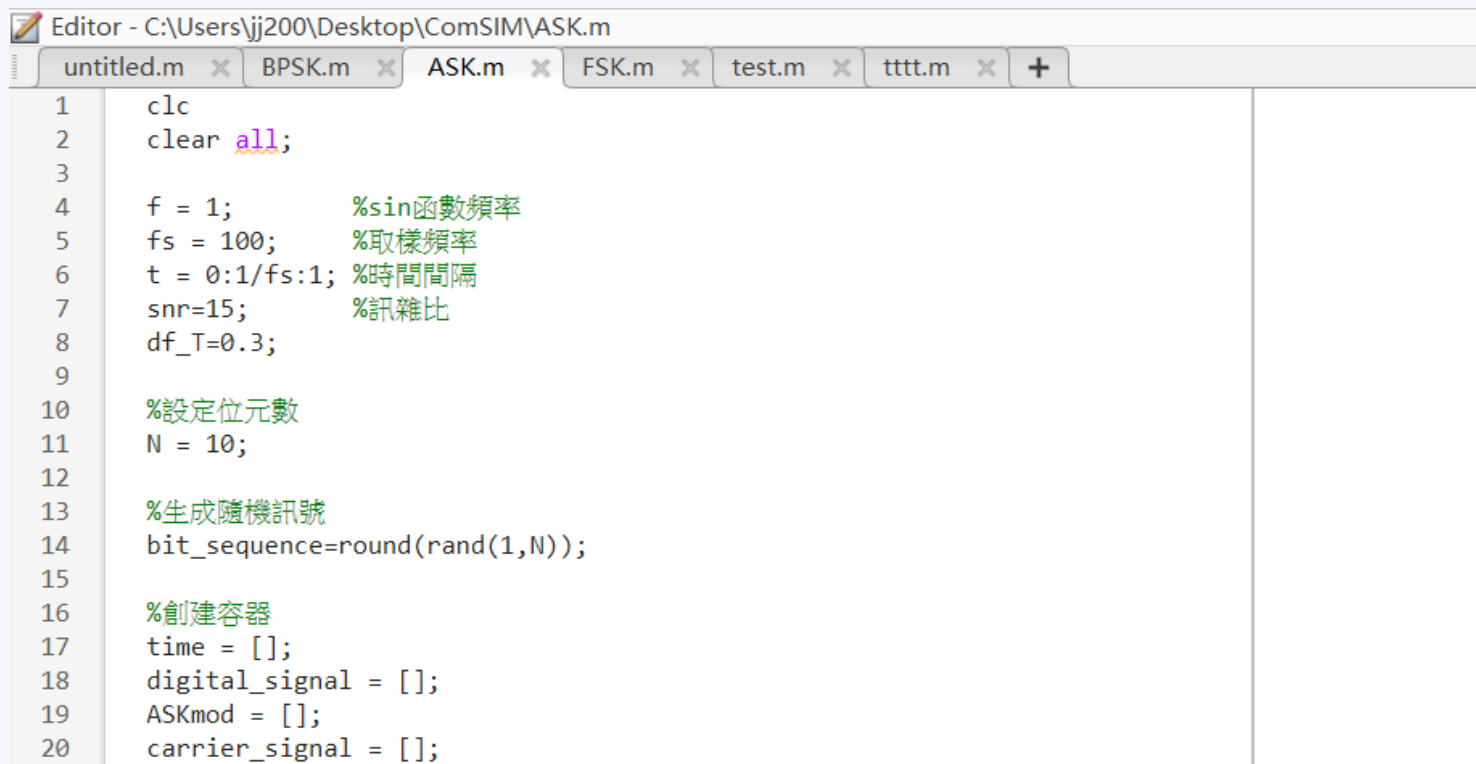


圖18. BPSK 系統

3-3.設計ASK系統



The image shows a MATLAB Editor window with the title bar "Editor - C:\Users\jj200\Desktop\ComSIM\ASK.m". The window contains several tabs: "untitled.m", "BPSK.m", "ASK.m", "FSK.m", "test.m", and "tttt.m". The "ASK.m" tab is active, displaying the following MATLAB code:

```
1  clc
2  clear all;
3
4  f = 1;      %sin函數頻率
5  fs = 100;   %取樣頻率
6  t = 0:1/fs:1; %時間間隔
7  snr=15;     %訊雜比
8  df_T=0.3;
9
10 %設定位元數
11 N = 10;
12
13 %生成隨機訊號
14 bit_sequence=round(rand(1,N));
15
16 %創建容器
17 time = [];
18 digital_signal = [];
19 ASKmod = [];
20 carrier_signal = [];
```

```

22 %產生訊號
23 for ii = 1:1:N %1:10
24
25     if bit_sequence(ii) == 0
26         bit = zeros(1,length(t));
27     else
28         bit = ones(1,length(t));
29     end
30     digital_signal = [digital_signal bit];
31
32     %產生ASK訊號
33     if bit_sequence(ii) == 0
34         bit = zeros(1,length(t));
35     else
36         bit = sin(2*pi*f*t);
37     end
38     ASKmod = [ASKmod bit];
39
40     %生成載波
41     carrier = sin(2*pi*f*t); % f=1 % t=0:1/fs:1
42     carrier_signal = [carrier_signal carrier];
43     time = [time t]; %時間間隔 1/fs =0.01
44     t = t + 1;
45 end
46
47 %加入AWGN Noise
48 ASK_AWGN=awgn(ASKmod,snr);
49
50 %Channel fading coefficients
51 K=exp(-j*2*pi*df_T);
52 ASK_off=ASKmod*K;
53
54 %頻偏後加入AWGN Noise
55 ASK_offset=awgn(ASK_off,snr);
56
57 %計算理論BER
58 BER_ASK=1/2*erfc(sqrt(snr/2));
59
60 % 計算加入雜訊之BER
61 BER_ASK_AWGN=abs(ASK_AWGN-ASKmod)/abs(ASKmod);
62
63 % 計算頻偏後加入雜訊之BER
64 BER_ASK_offset=abs(ASK_offset-ASKmod)/abs(ASKmod);

```

```
66 %繪圖
67 subplot(5,1,1);
68 plot(time,ASKmod,'r','linewidth',2);
69 title({'理想 ASK 調變訊號', ['BER=', num2str(BER_ASK)] })
70 grid on;
71 axis tight;
72
73 subplot(5,1,2);
74 plot(time,ASK_AWGN,'r','linewidth',2);
75 title({'ASK+AWGN 調變訊號', ['BER=', num2str(BER_ASK_AWGN)] })
76 grid on;
77 axis tight;
78
79 subplot(5,1,3);
80 plot(time,ASK_offset,'r','linewidth',2);
81 title({'頻偏之 ASK+AWGN 調變訊號', ['BER=', num2str(BER_ASK_offset)] })
82 grid on;
83 axis tight;
84
85 subplot(5,1,4);
86 plot(time,digital_signal,'linewidth',2);% linewidth:調整線條粗度
87 title('位元序列')
88 grid on;
89 axis([0 time(end) -0.5 1.5]);
90
91 subplot(5,1,5);
92 plot(time,carrier_signal,'linewidth',2);
93 title('載波訊號')
94 grid on;
95 axis tight;
```

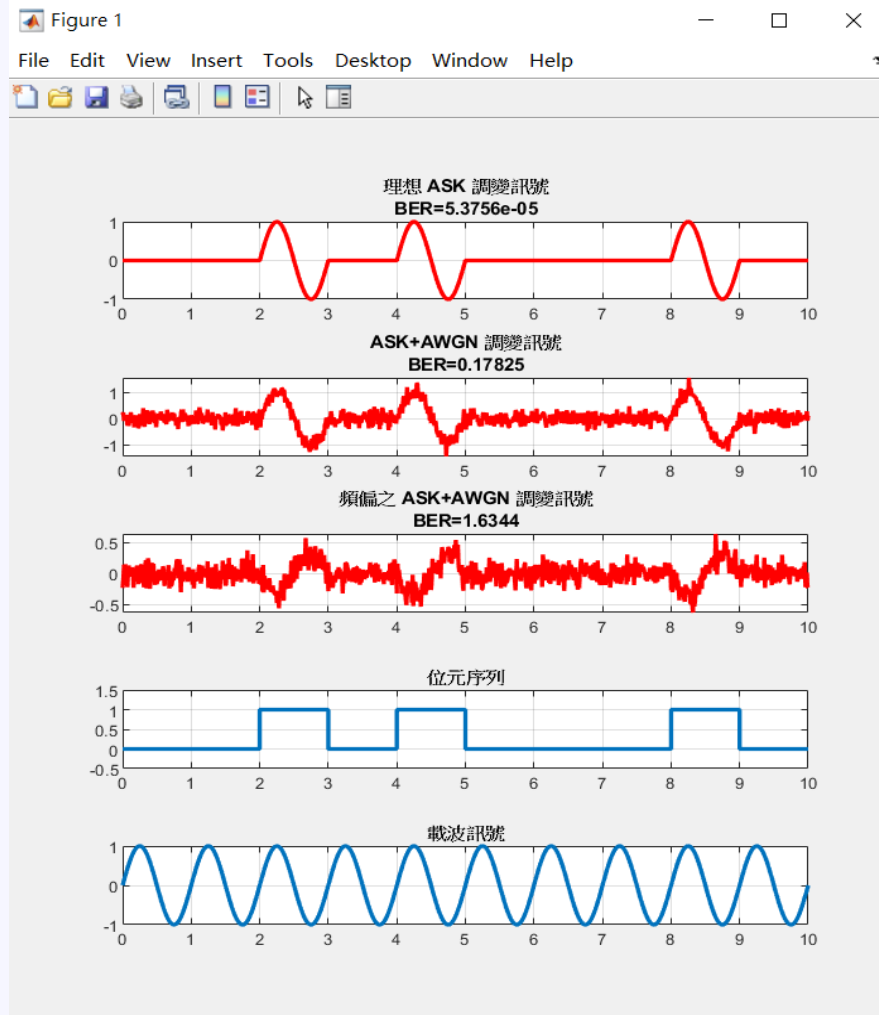
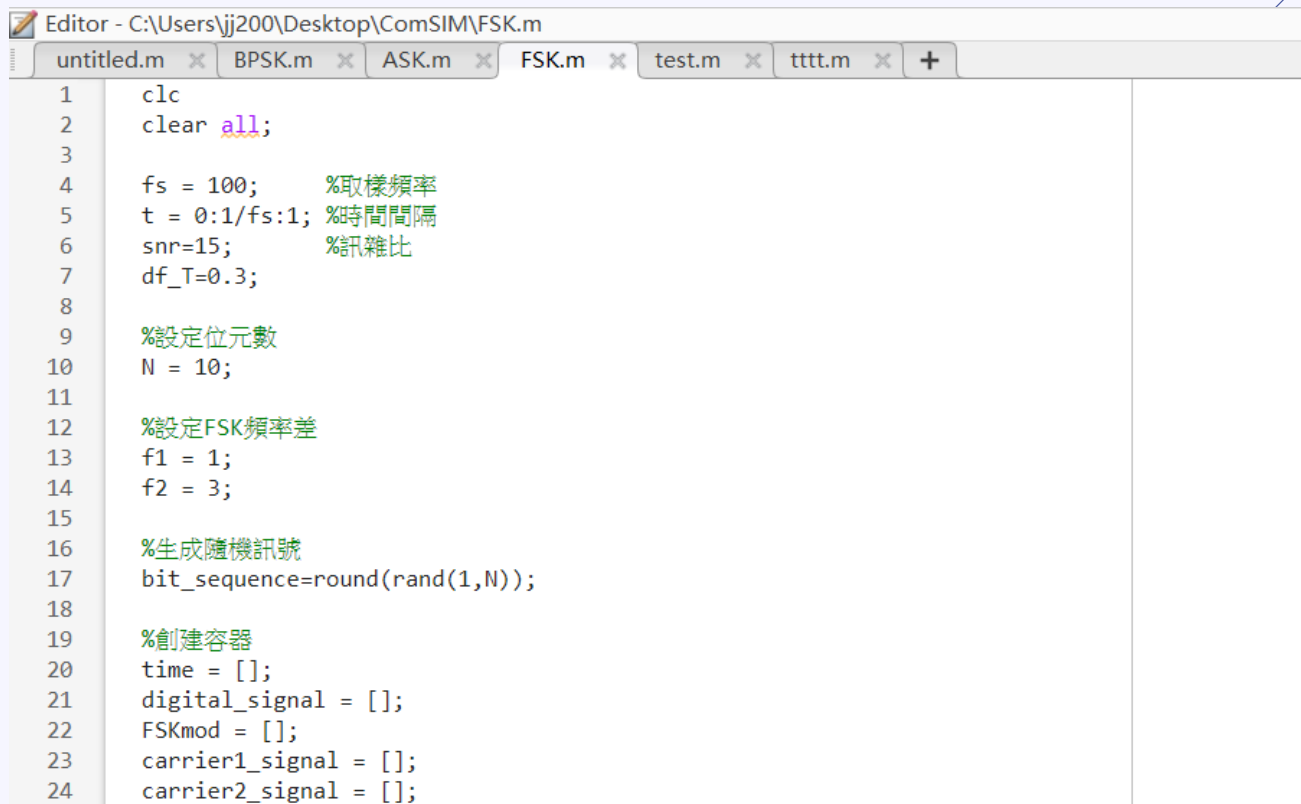


圖19. ASK 系統

3-4.設計FSK系統



The image shows a MATLAB Editor window with the title bar "Editor - C:\Users\jj200\Desktop\ComSIM\FSK.m". The window contains a script with 24 lines of MATLAB code. The code defines parameters for an FSK system, including sampling frequency, time interval, SNR, frequency difference, number of bits, and FSK frequencies. It also generates a random bit sequence and initializes arrays for time, digital signal, FSK modulation, and carrier signals.

```
1  clc
2  clear all;
3
4  fs = 100;    %取樣頻率
5  t = 0:1/fs:1; %時間間隔
6  snr=15;     %訊雜比
7  df_T=0.3;
8
9  %設定位元數
10 N = 10;
11
12 %設定FSK頻率差
13 f1 = 1;
14 f2 = 3;
15
16 %生成隨機訊號
17 bit_sequence=round(rand(1,N));
18
19 %創建容器
20 time = [];
21 digital_signal = [];
22 FSKmod = [];
23 carrier1_signal = [];
24 carrier2_signal = [];
```

```

26 %產生訊號
27 for ii = 1:1:N %1:10
28
29     if bit_sequence(ii) == 0
30         bit = zeros(1,length(t));
31     else
32         bit = ones(1,length(t));
33     end
34     digital_signal = [digital_signal bit];
35
36     %產生FSK訊號
37     if bit_sequence(ii) == 0
38         bit = sin(2*pi*f1*t);
39     elseif bit_sequence(ii) == 1
40         bit = sin(2*pi*f2*t);
41     end
42     FSKmod = [FSKmod bit];
43
44     %生成載波1
45     carrier1 = sin(2*pi*f1*t); % f1=1 % t=0:1/fs:1
46     carrier1_signal = [carrier1_signal carrier1];
47
48     %生成載波2
49     carrier2 = sin(2*pi*f2*t); % f2=2 % t=0:1/fs:1
50     carrier2_signal = [carrier2_signal carrier2];
51
52     time = [time t]; %時間間隔 1/fs =0.01
53     t = t + 1;
54 end
55
56 %加入AWGN Noise
57 FSK_AWGN=awgn(FSKmod,snr);
58
59 %Channel fading coefficients
60 K=exp(-j*2*pi*df_T);
61 FSK_offset=FSKmod*K;
62
63 %頻偏後加入AWGN Noise
64 FSK_offset=awgn(FSK_offset,snr);
65
66 %計算理論BER
67 BER_FSK=1/2*erfc(sqrt(snr/2));
68
69 % 計算加入雜訊之BER
70 BER_FSK_AWGN=abs(FSK_AWGN-FSKmod)/abs(FSKmod);
71
72 % 計算頻偏後加入雜訊之BER
73 BER_FSK_offset=abs(FSK_offset-FSKmod)/abs(FSKmod);

```

```
75 %繪圖
76 subplot(6,1,1);
77 plot(time,FSKmod,'r','linewidth',2);
78 title({'理想 FSK 調變訊號',[ 'BER=',num2str(BER_FSK)] })
79 grid on;
80 axis tight;
81
82 subplot(6,1,2);
83 plot(time,FSK_AWGN,'r','linewidth',2);
84 title({'FSK+AWGN 調變訊號',[ 'BER=',num2str(BER_FSK_AWGN)] })
85 grid on;
86 axis tight;
87
88 subplot(6,1,3);
89 plot(time,FSK_AWGN,'r','linewidth',2);
90 title({'頻偏之 FSK+AWGN 調變訊號',[ 'BER=',num2str(BER_FSK_offset)] })
91 grid on;
92 axis tight;
93
94 subplot(6,1,4);
95 plot(time,digital_signal,'linewidth',2);% linewidth:調整線條粗度
96 title('位元序列')
97 grid on;
98 axis([0 time(end) -0.5 1.5]);
99
100 subplot(6,1,5);
101 plot(time,carrier1_signal,'linewidth',2);
102 title('載波訊號1')
103 grid on;
104 axis tight;
105
106 subplot(6,1,6);
107 plot(time,carrier2_signal,'linewidth',2);
108 title('載波訊號2')
109 grid on;
110 axis tight;
```

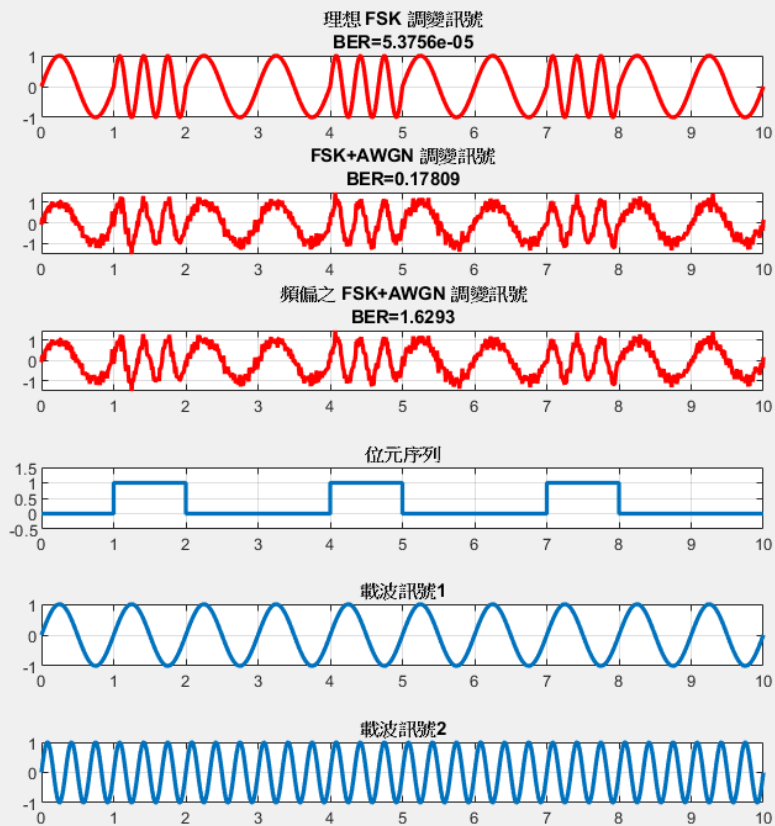
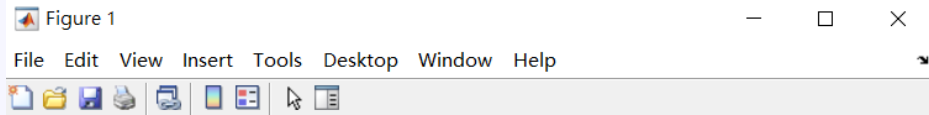


圖20. FSK 系統

實驗心得

• 實驗一

透過實驗一讓我學習到如何設計一個PN序列的LFSR系統與產生OVSF架構，也藉由教材的引導了解其他如M-sequence 與 4G PSS系統之概要，以及 ZC sequence和自相關的特性。

• 實驗二

透過分析Linear Convolution & Circular Convolution，讓我了解他們的卷積特性，並藉由在資料閱讀的過程中，找出了使兩者相等之所需條件，此外也利用程式驗證了Parseval's Theorem於FFT前後能量相等之關係，並利用Vandermonde 矩陣設計出DFT矩陣。

• 實驗三、實驗四

這個實驗主要是透過每個禮拜不同的主題，逐漸使系統增加更多的功能與參數，從一開始設計出單純的QPSK、BPSK、ASK、FSK 等系統，到後面比較GrayCode、加入AWGN雜訊、頻率偏移、BER分析等，使系統越來越龐大複雜，因此透過這個實驗也讓我學習了如何細心簡化程式，梳理好整體架構後查詢相關理論Debug，並利用不同參數設計出美觀的比較圖，也使我更了解到各個系統間之關係以及雜訊與頻率偏移對於系統之影響。



Thanks