OFDM with a Fading Channel

Table of Contents

Housekeeping and Constant Assignment

```
close all
clc

for L=[16 64 256]  %size of IFFT/FFT

    if L==16
        m=4;  %m = log2(M) is the number bits per symbol or M = 2^m
        M=2^m;    %number of symbols in M-ary modulation
    elseif L==64
        m=6;    %m = log2(M) is the number bits per symbol or M = 2^m
        M=2^m;    %number of symbols in M-ary modulation
    else
        m=8;    %m = log2(M) is the number bits per symbol or M = 2^m
        M=2^m;    %number of symbols in M-ary modulation
    end

N=200;  %number of OFDM symbols
```

(a) Generate Input Bit Sequence

Generate a random sequence bk of 1s and 0s with equal probability of length m×L×N. Suggested values are: m = 2; L = 16 (64 and 256 are optional); and N = 200 or more.

```
bits_bk =randi([0 1],[N*L*m 1]);    %Input Bit Sequence
```

(b) Convert Bit Sequence to Bit Group Sequence

```
Let us consider QPSK, i.e., m = 2 or M = 4. A higher order modulation is
an option. Convert groups of m bits mk into a sequence of unsigned
decimal values.
```

```
bitsTo_M_Vectors_mk = reshape(bits_bk,length(bits_bk)/m,m);     %Bit Group
DecimalVector = bi2de(bitsTo_M_Vectors_mk);
```

(c) Convert Bit Group Sequence into Constellation Symbol Group

```
Use the constellation diagram below to map mk into constellation symbol
sequence Xk. These are complex values (I-Q data).
```

```
Symbols_Xk = qammod(DecimalVector,L,0,'gray');  %Constellation Symbol Group
```

(d) Apply Constellation Symbol Groups to IFFT

```
Take a block of size L constellation symbols and apply the IFFT
algorithm. Repeat this N times. These are complex valued too.
```

```
SymbolsIFFT = ifft(Symbols_Xk);
```

(e) Calculate Gaussian Noise and Rayleigh Fading for Signal

```
Simulate the channel as appropriate for each scenario. For AWGN, add
(I-Q) white Gaussian noise with zero mean; suggested standard deviation
values are ? = 0, 0.02, 0.08. The fading cannel can be realized in a
number of ways with Rayleigh being the simplest. Here are two models
to consider (the second is optional).
```

```
sigmaValues = [0.0, 0.02, 0.08];    %Standard Deviation
randQ=randn(length(Symbols_Xk),1)*i;    %Creates random variations of Q
randI=randn(length(Symbols_Xk),1);      %Creates random variations of I
Noise=(randQ+randI)*sigmaValues;        %Imposes Noise to the signal vector
% Add Stanford University Interium Model Fading to
%SUI channel   y(n)=x(n)+0.5x(n-1)+0.25x(n-2)

% Add Rayleigh Fading to the Signal
%  Rayleigh (statistical model). The effect of fading is realized as a
% scalar h, given by h= sqrt(abs(wI+jwQ)) where wI and wQ are independent
% white Gaussian random variables with unit variance. Hint: compute a
% new h value for each message symbol period.
h=sqrt(abs(Noise));
```

(ea) Fading Scenario 1 SingleCarrier Modulation,fading Channel

```
SymbolsOut_rk_SingleCarrier(1)=Symbols_Xk(1);
SymbolsOut_rk_SingleCarrier(2)=Symbols_Xk(2)+0.5* ...
    SymbolsOut_rk_SingleCarrier(1);
for i=3:N*L
    SymbolsOut_rk_SingleCarrier(i)=Symbols_Xk(i)+0.5*Symbols_Xk(i-1)+ ...
        0.25*Symbols_Xk(i-2);
end
```

(eb) Fading Scenario 2 Multicarrier Modulation, fading channel

```
SymbolsOut_rk_woNoise(1)=SymbolsIFFT(1);
SymbolsOut_rk_woNoise(2)=SymbolsIFFT(2)+0.5*SymbolsIFFT(1);
for i=3:N*L
    SymbolsOut_rk_woNoise(i)=SymbolsIFFT(i)+0.5*SymbolsIFFT(i-1)+ ...
        0.25*Symbols_Xk(i-2);
end
```

(ec) Noise Scenario 3 MultiCarrier Modulation, AWGN channel

```
SymbolsOut_rk(:,1)=SymbolsIFFT+Noise(:,1);
SymbolsOut_rk(:,2)=SymbolsIFFT+Noise(:,2);
SymbolsOut_rk(:,3)=SymbolsIFFT+Noise(:,3);
```

(f) Use FFT to Recover Constellation Symbol

(fa) N/A (fb) FFT Scenario 2 Multicarrier Modulation, fading channel

```
SymbolsInFFT_Rayleigh = fft(SymbolsOut_rk_woNoise);

% (fc) FFT Scenario 3 MultiCarrier Modulation, AWGN channel
SymbolsInFFT_AWGM(:,1) = fft(SymbolsOut_rk(:,1));
SymbolsInFFT_AWGM(:,2) = fft(SymbolsOut_rk(:,2));
SymbolsInFFT_AWGM(:,3) = fft(SymbolsOut_rk(:,3));
```

(g) Implement Demapping to Return Received Symbols to Bit Stream

(ga) Demap Scenario 1 SingleCarrier Modulation,fading Channel

```
SymbolsIn_dk_Single(1,:) = qamdemod(SymbolsOut_rk_SingleCarrier,L,0,...
```

```
    'gray');

% (gb) Demap Scenario 2 Multicarrier Modulation,Raleigh fading channel
SymbolsIn_dk_Rayleigh(1,:) = qamdemod(SymbolsInFFT_Rayleigh,L,0,...
    'gray');

% (gc) Demap Scenario 3 MultiCarrier Modulation, AWGN channel
SymbolsIn_dk(:,1) = qamdemod(SymbolsInFFT_AWGM(:,1),L,0,'gray');
SymbolsIn_dk(:,2) = qamdemod(SymbolsInFFT_AWGM(:,2),L,0,'gray');
SymbolsIn_dk(:,3) = qamdemod(SymbolsInFFT_AWGM(:,3),L,0,'gray');
```

(h) Convert From Integer Symbols to Binary

```
%Rayleigh-Single Carrier
DecimalVector_To_Binary_ck_Single = de2bi(SymbolsIn_dk_Single(:),m);
bitsIn_ck_Single=reshape(DecimalVector_To_Binary_ck_Single,[N*L*m 1]);

%Rayleigh-MultiCarrier
DecimalVector_To_Binary_ck_Rayleigh = de2bi(SymbolsIn_dk_Rayleigh,m);
bitsIn_ck_Rayleigh=reshape(DecimalVector_To_Binary_ck_Rayleigh,[N*L*m 1]);

%AWGM-MultiCarrier
DecimalVector_To_Binary_ck(:,1:m) = de2bi(SymbolsIn_dk(:,1),m);
DecimalVector_To_Binary_ck(:,m+1:2*m) = de2bi(SymbolsIn_dk(:,2),m);
DecimalVector_To_Binary_ck(:,2*m+1:3*m) = de2bi(SymbolsIn_dk(:,3),m);

bitsIn_ck(:,1)=reshape(DecimalVector_To_Binary_ck(:,1:m),[N*L*m 1]);
bitsIn_ck(:,2)=reshape(DecimalVector_To_Binary_ck(:,m+1:2*m),[N*L*m 1]);
bitsIn_ck(:,3)=reshape(DecimalVector_To_Binary_ck(:,2*m+1:3*m),[N*L*m 1]);
```

(i) Calculate Bit Error Rates and display

```
fprintf('\n%d Bits Per Symbol over %d samples resulted in:\n',L,N)
[numErrors,ber]=biterr(bits_bk,bitsIn_ck_Single);
fprintf('Single Carrier Stanford Model had a bit error rate of:\n')
fprintf('%5.2e(%d errors).\n', ber,numErrors)
[numErrors,ber]=biterr(bits_bk,bitsIn_ck_Rayleigh);
fprintf('Multiple Carrier Stanford Model had a bit error rate of:\n')
fprintf('%5.2e(%d errors).\n', ber,numErrors)
[numErrors,ber]=biterr(bits_bk,bitsIn_ck(:,1));
fprintf('Multiple Carrier AWGN had a bit error rate of:\n')
fprintf('%5.2e(%d errors) with Sigma=0.\n', ber,numErrors)
[numErrors,ber]=biterr(bits_bk,bitsIn_ck(:,2));
fprintf('Multiple Carrier AWGN had a bit error rate of:\n')
fprintf('%5.2e(%d errors) with Sigma=0.02.\n' , ber,numErrors)
[numErrors,ber]=biterr(bits_bk,bitsIn_ck(:,3));
fprintf('Multiple Carrier AWGN had a bit error rate of:\n')
fprintf('%5.2e(%d errors) with Sigma=0.08.\n', ber,numErrors)
```

```
16 Bits Per Symbol over 200 samples resulted in:
Single Carrier Stanford Model had a bit error rate of:
```

1.87e-01(2390 errors).
Multiple Carrier Stanford Model had a bit error rate of:
4.90e-01(6269 errors).
Multiple Carrier AWGN had a bit error rate of:
0.00e+00(0 errors) with Sigma=0.
Multiple Carrier AWGN had a bit error rate of:
1.40e-01(1795 errors) with Sigma=0.02.
Multiple Carrier AWGN had a bit error rate of:
4.02e-01(5150 errors) with Sigma=0.08.

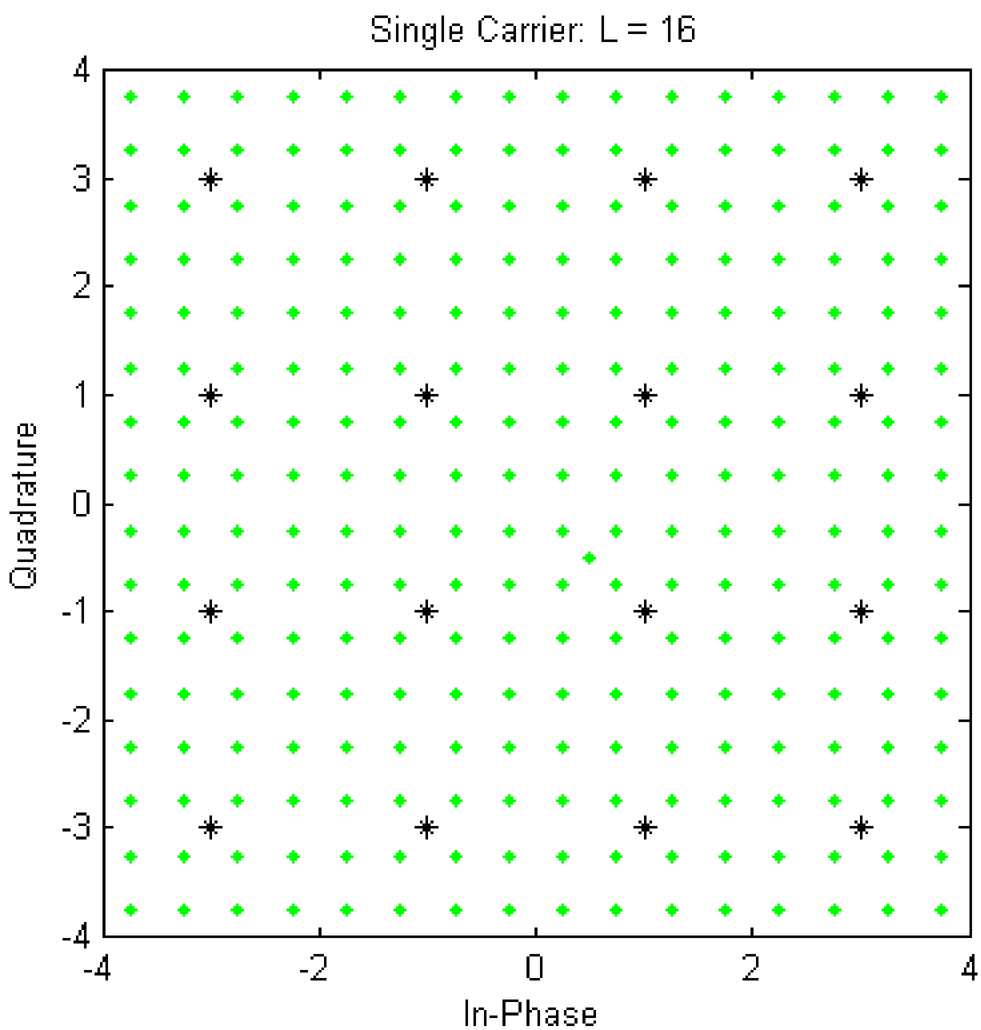
64 Bits Per Symbol over 200 samples resulted in:
Single Carrier Stanford Model had a bit error rate of:
2.87e-01(22068 errors).
Multiple Carrier Stanford Model had a bit error rate of:
4.93e-01(37864 errors).
Multiple Carrier AWGN had a bit error rate of:
0.00e+00(0 errors) with Sigma=0.
Multiple Carrier AWGN had a bit error rate of:
2.38e-01(18284 errors) with Sigma=0.02.
Multiple Carrier AWGN had a bit error rate of:
4.34e-01(33341 errors) with Sigma=0.08.

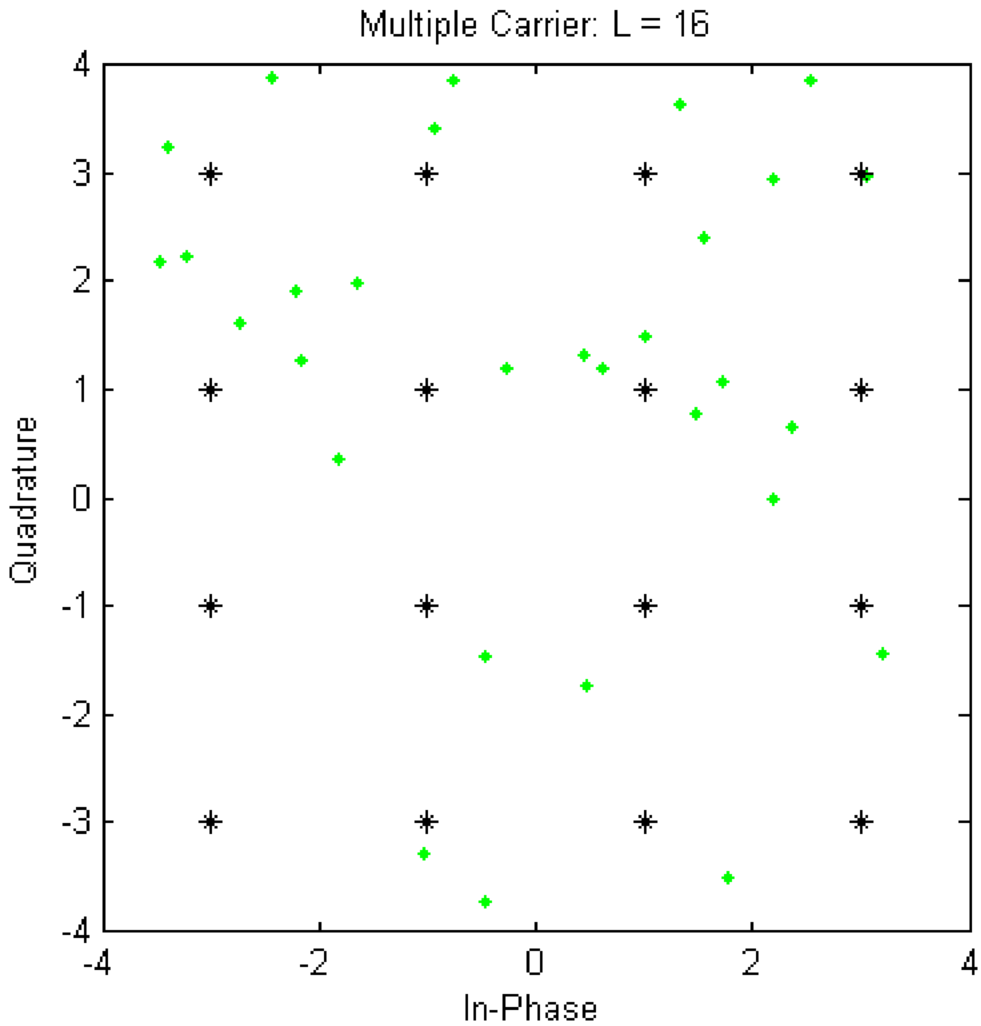
256 Bits Per Symbol over 200 samples resulted in:
Single Carrier Stanford Model had a bit error rate of:
3.42e-01(139914 errors).
Multiple Carrier Stanford Model had a bit error rate of:
4.99e-01(204315 errors).
Multiple Carrier AWGN had a bit error rate of:
0.00e+00(0 errors) with Sigma=0.
Multiple Carrier AWGN had a bit error rate of:
3.05e-01(125001 errors) with Sigma=0.02.
Multiple Carrier AWGN had a bit error rate of:
4.51e-01(184823 errors) with Sigma=0.08.

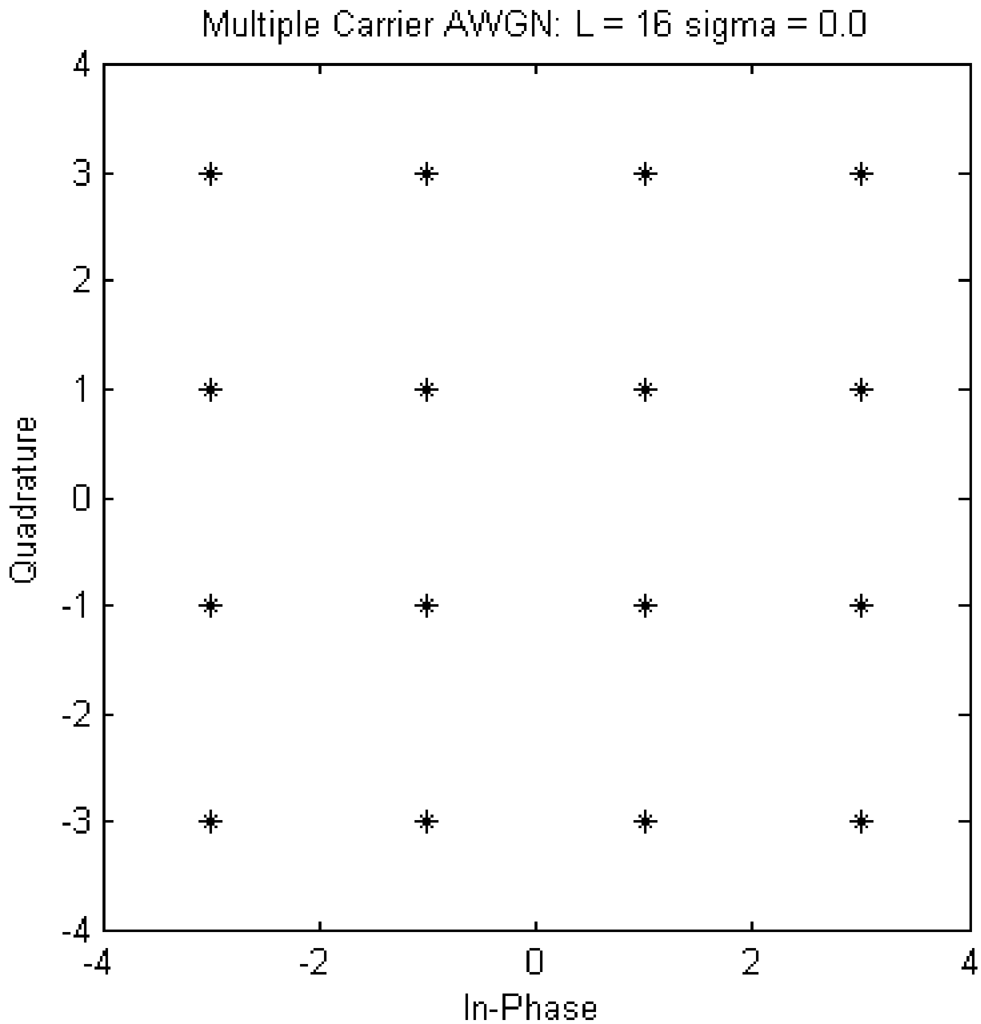
(j) Graphs

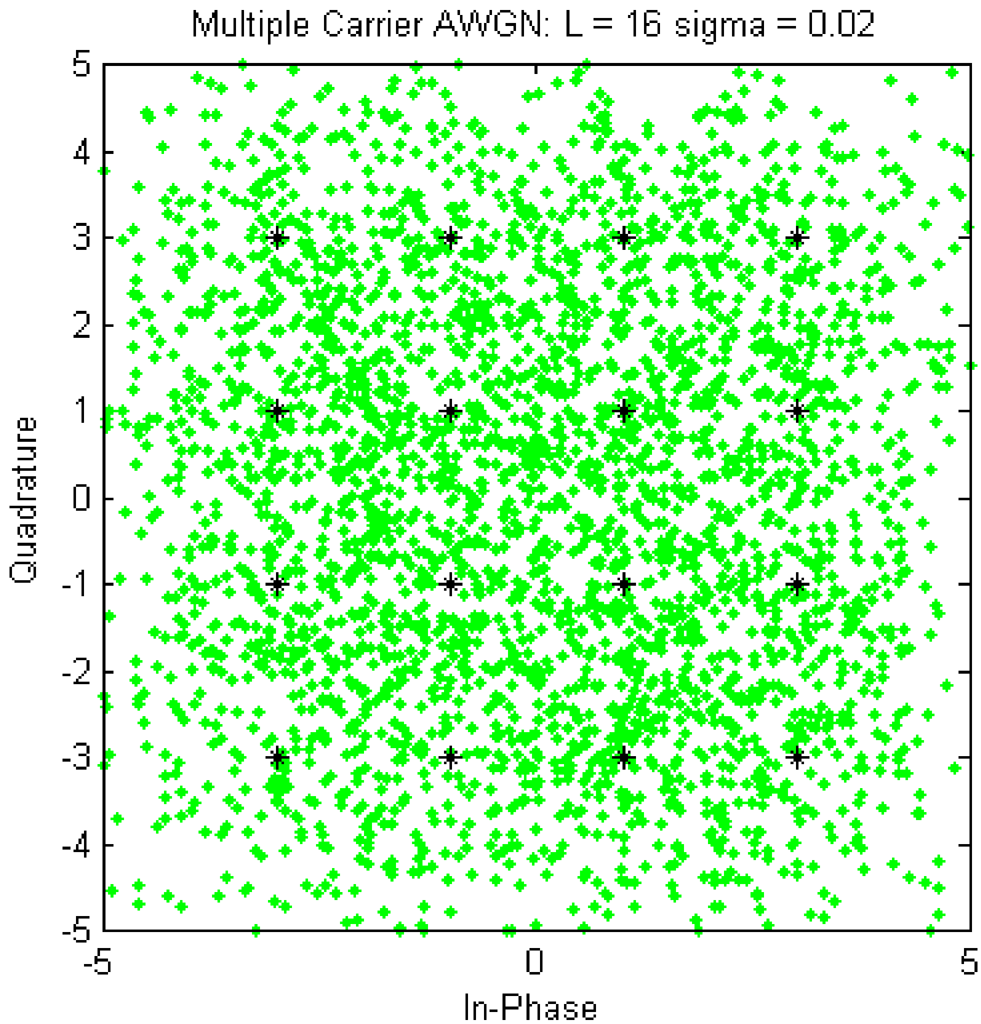
```
splotfig1=scatterplot(SymbolsOut_rk_SingleCarrier,1,0,'g.');
hold on
scatterplot(Symbols_Xk,1,0,'k*',splotfig1);
title(strcat('Single Carrier: L = ', {' '},num2str(L)))
axis([-m m -m m])

splotfig4=scatterplot(SymbolsInFFT_Rayleigh,1,0,'g.');
hold on
scatterplot(Symbols_Xk,1,0,'k*',splotfig4);
title(strcat('Multiple Carrier: L = ', {' '},num2str(L)))
axis([-m m -m m])

splotfig7=scatterplot(SymbolsInFFT_AWGM(:,1),1,0,'g.');
hold on
scatterplot(Symbols_Xk,1,0,'k*',splotfig7);
```

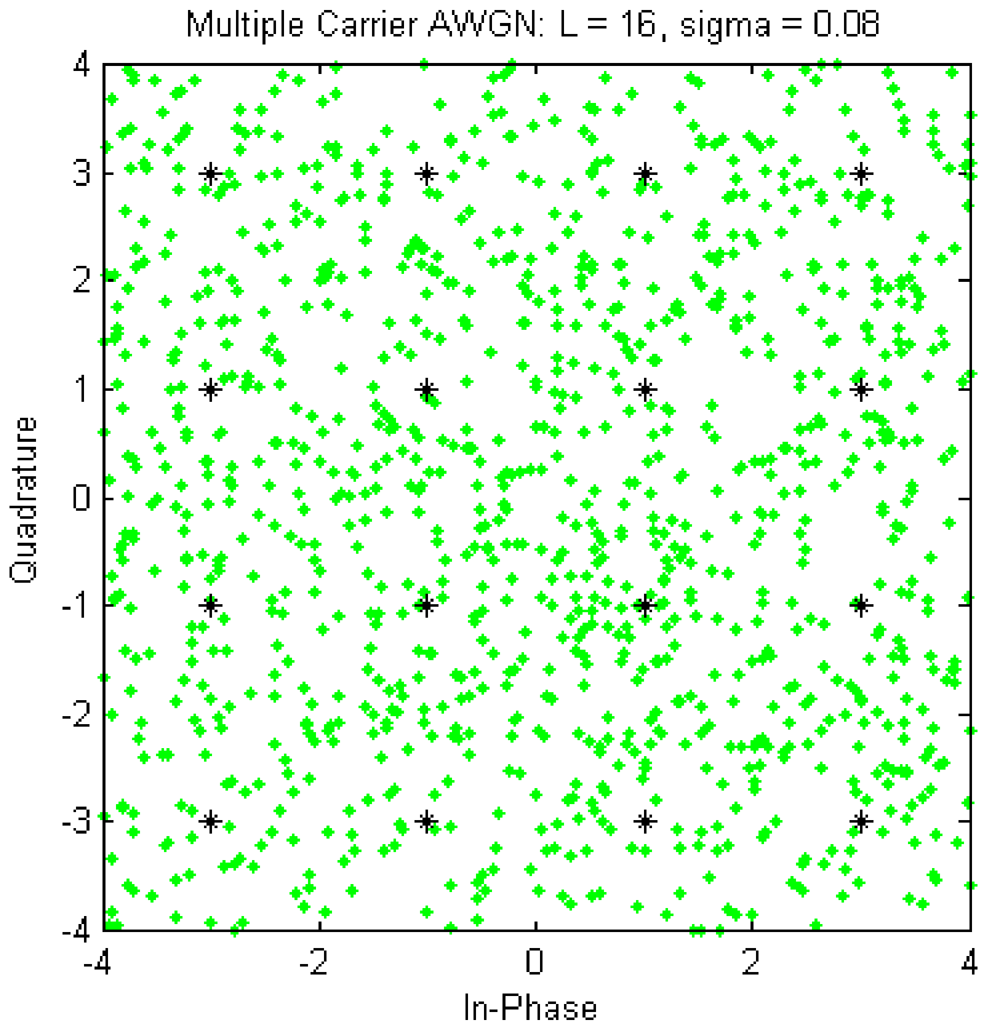
```
title(strcat('Multiple Carrier AWGN: L = ', {' '},num2str(L), ...
    ' sigma = 0.0'))
axis([-m m -m m])

splotfig8=scatterplot(SymbolsInFFT_AWGM(:,2),1,0,'g.');
hold on
scatterplot(Symbols_Xk,1,0,'k*',splotfig8);
title(strcat('Multiple Carrier AWGN: L = ', {' '},num2str(L), ...
    ' sigma = 0.02'))
axis([-m-1 m+1 -m-1 m+1])

splotfig9=scatterplot(SymbolsInFFT_AWGM(:,3),1,0,'g.');
hold on
scatterplot(Symbols_Xk,1,0,'k*',splotfig9);
title(strcat('Multiple Carrier AWGN: L =  ', {' '},num2str(L), ...
    ', sigma = 0.08'))
axis([-m m -m m])
```

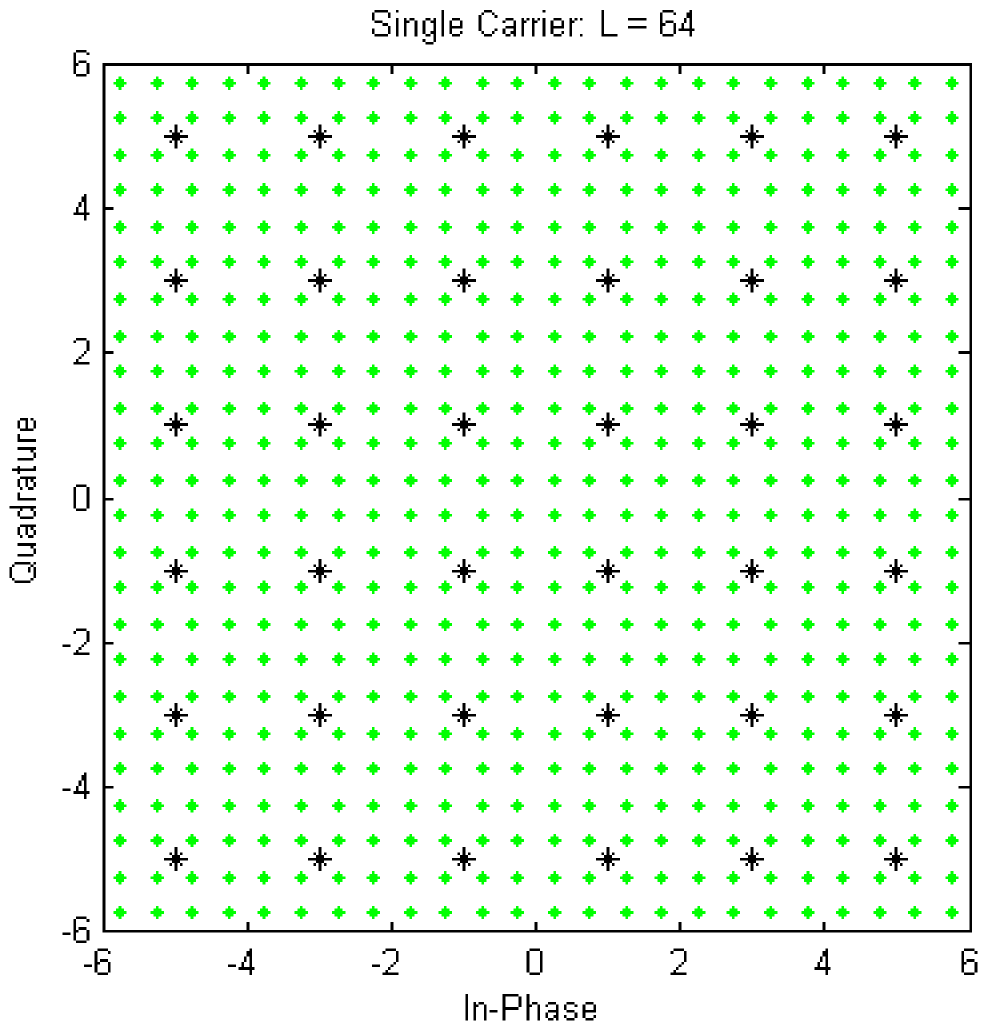


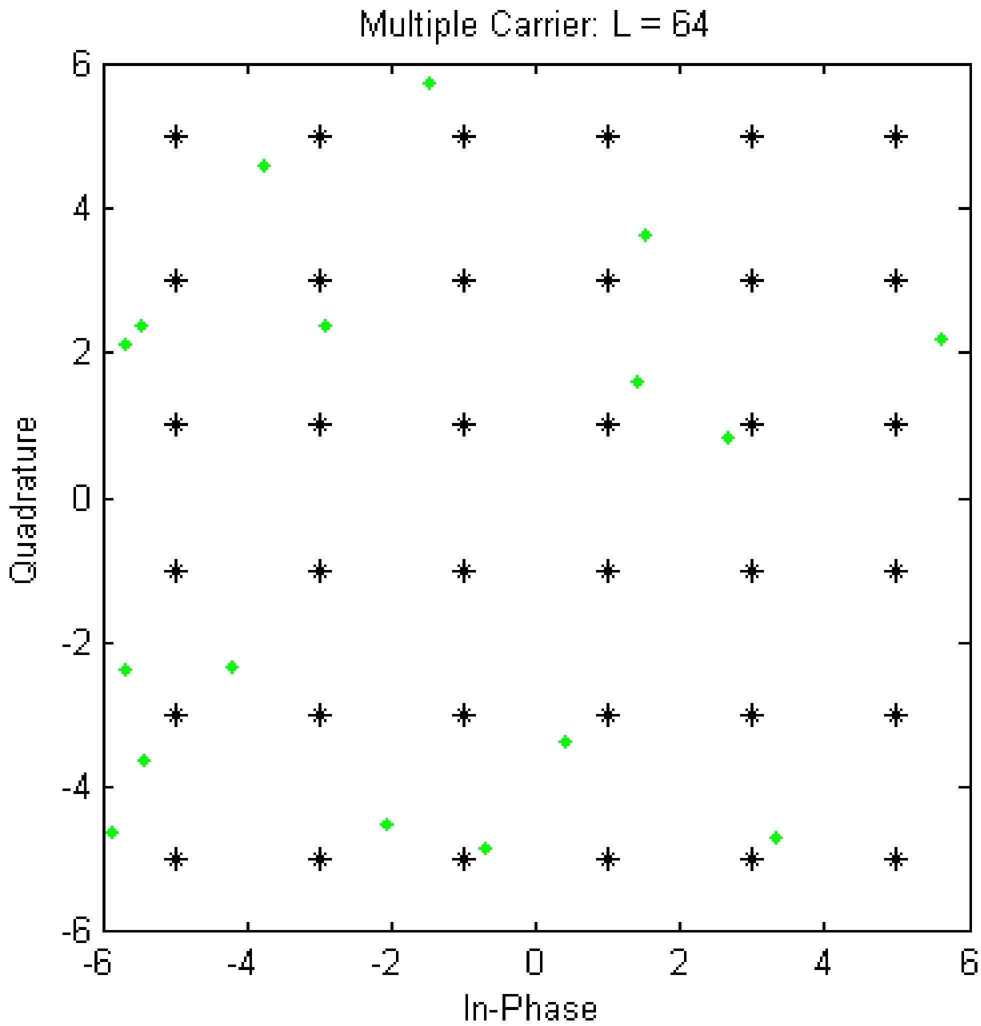

Multiple Carrier: L = 16
Quadrature
In-Phase
4
3
2
1
0
-1
-2
-3
-4
-4
-2
0
2
4

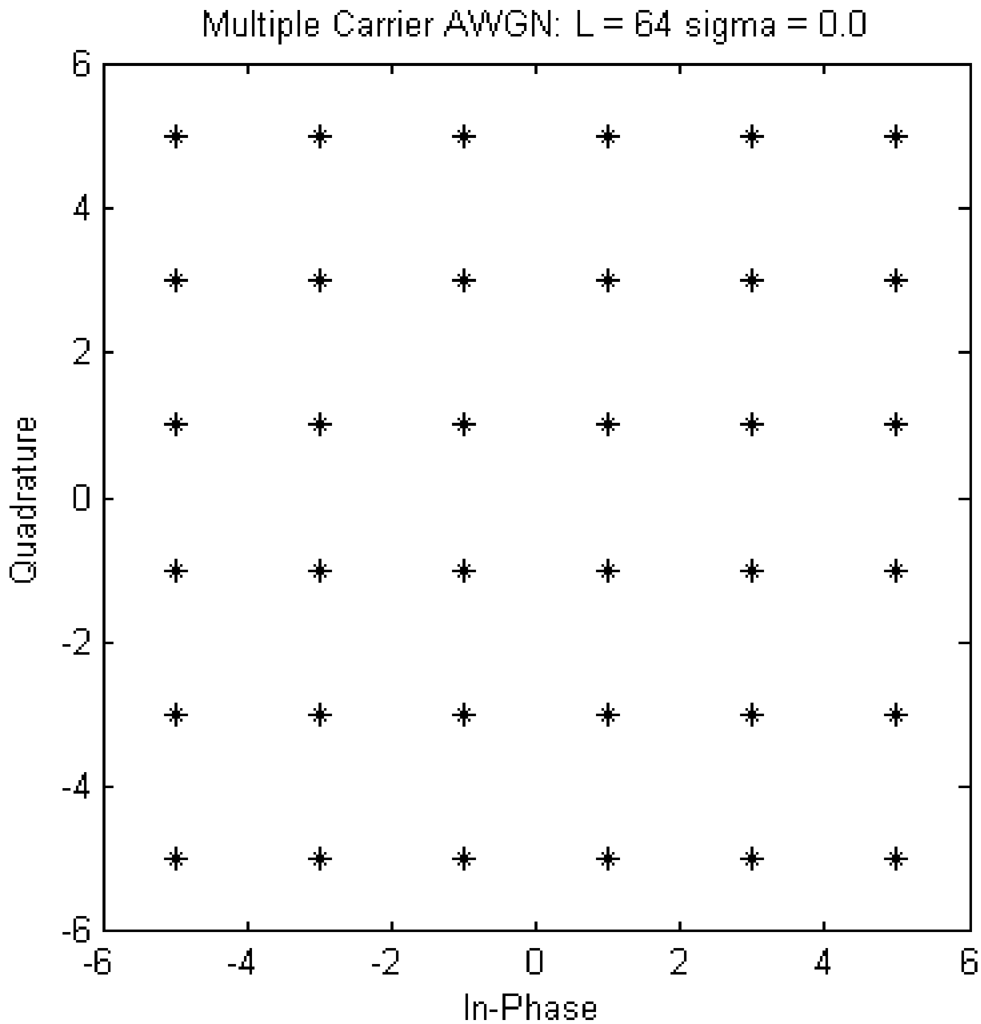

Multiple Carrier AWGN: L = 16 sigma = 0.0
4
3
2
1
0
-1
-2
-3
-4
Quadrature
-4
-2
0
2
4
In-Phase

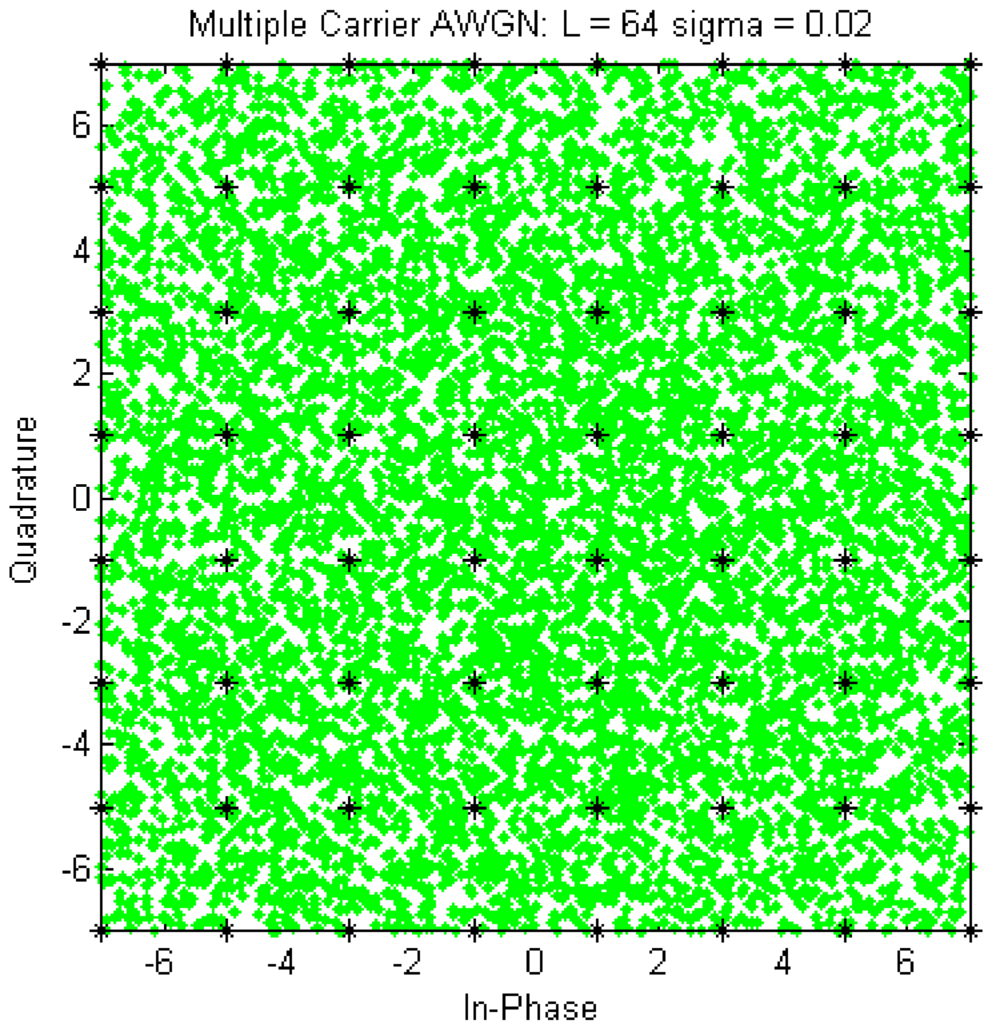

Multiple Carrier AWGN: L = 16 sigma = 0.02
Quadrature
In-Phase
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
0
5

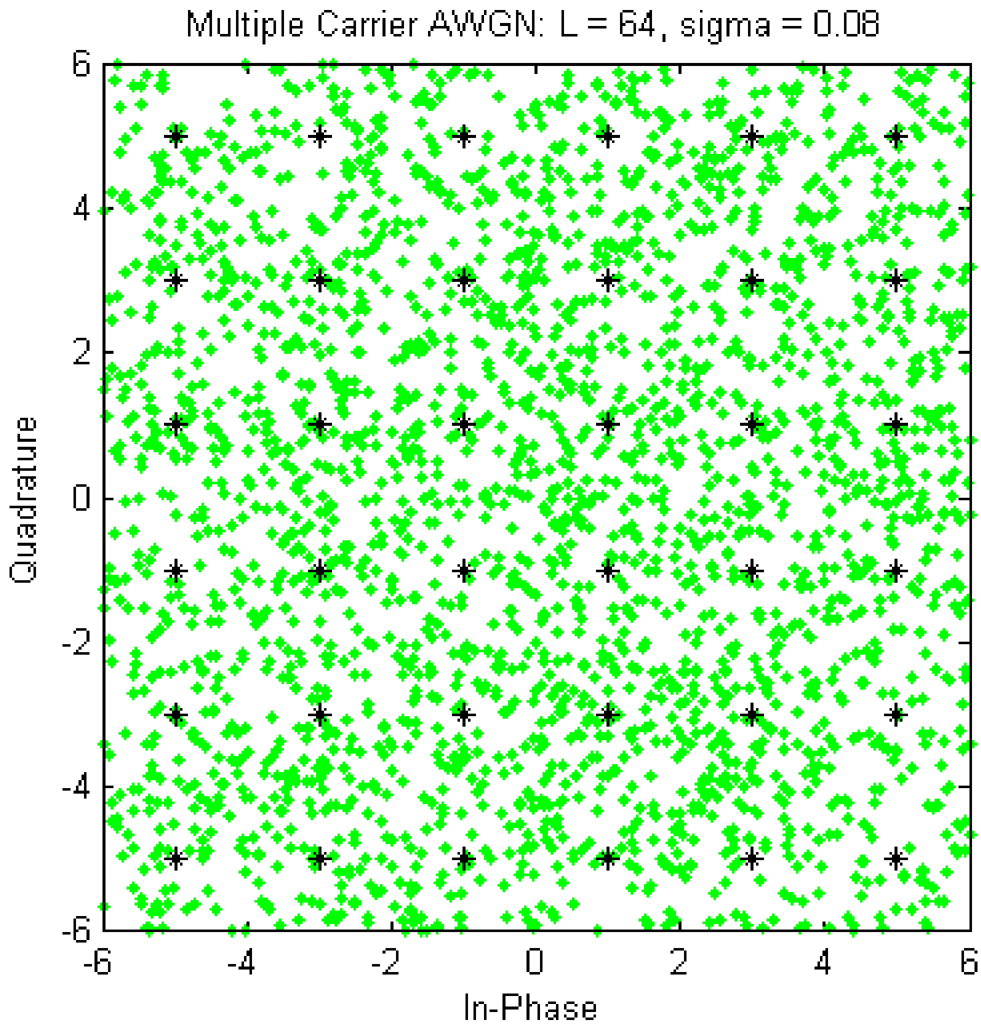

Multiple Carrier AWGN: L = 16, sigma = 0.08
4
3
2
1
0
-1
-2
-3
-4
Quadrature
-4
-2
0
2
4
In-Phase

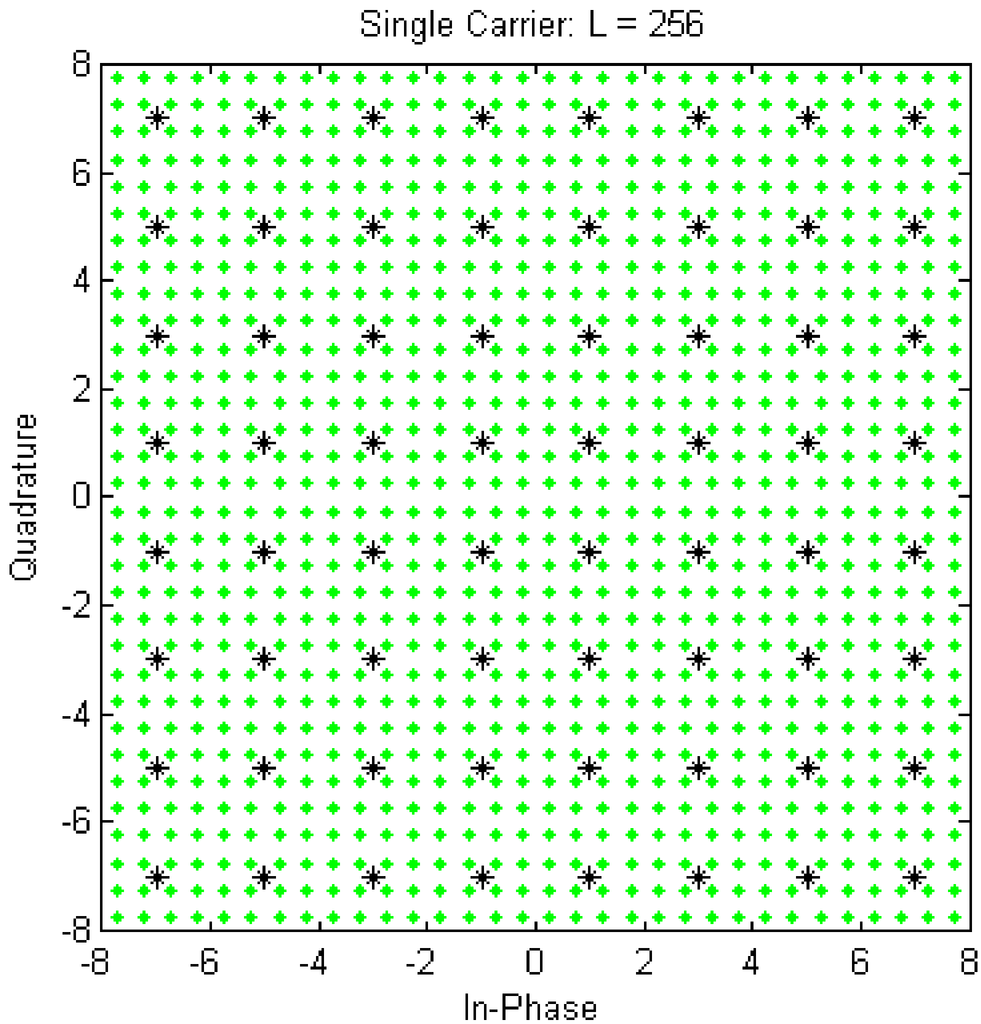

Single Carrier: L = 64
Quadrature
In-Phase
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

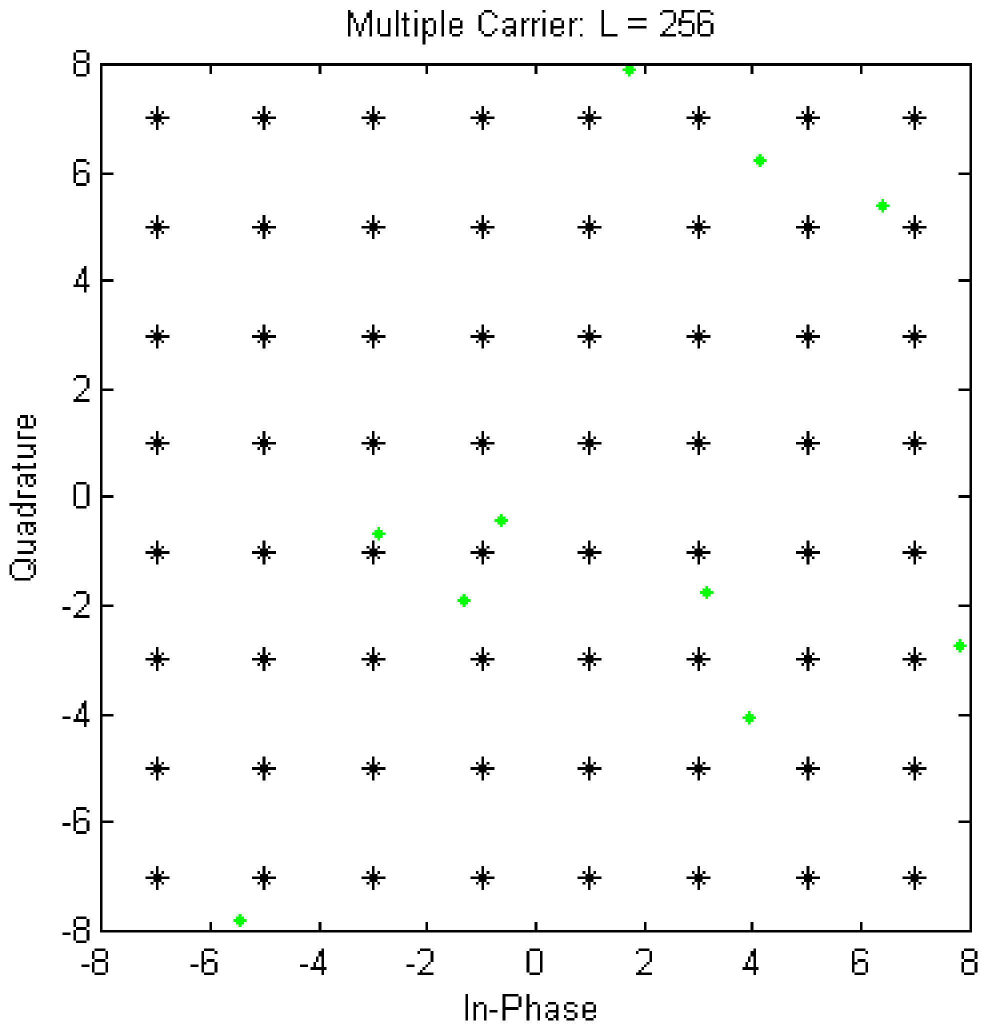

Multiple Carrier: L = 64
Quadrature
In-Phase
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

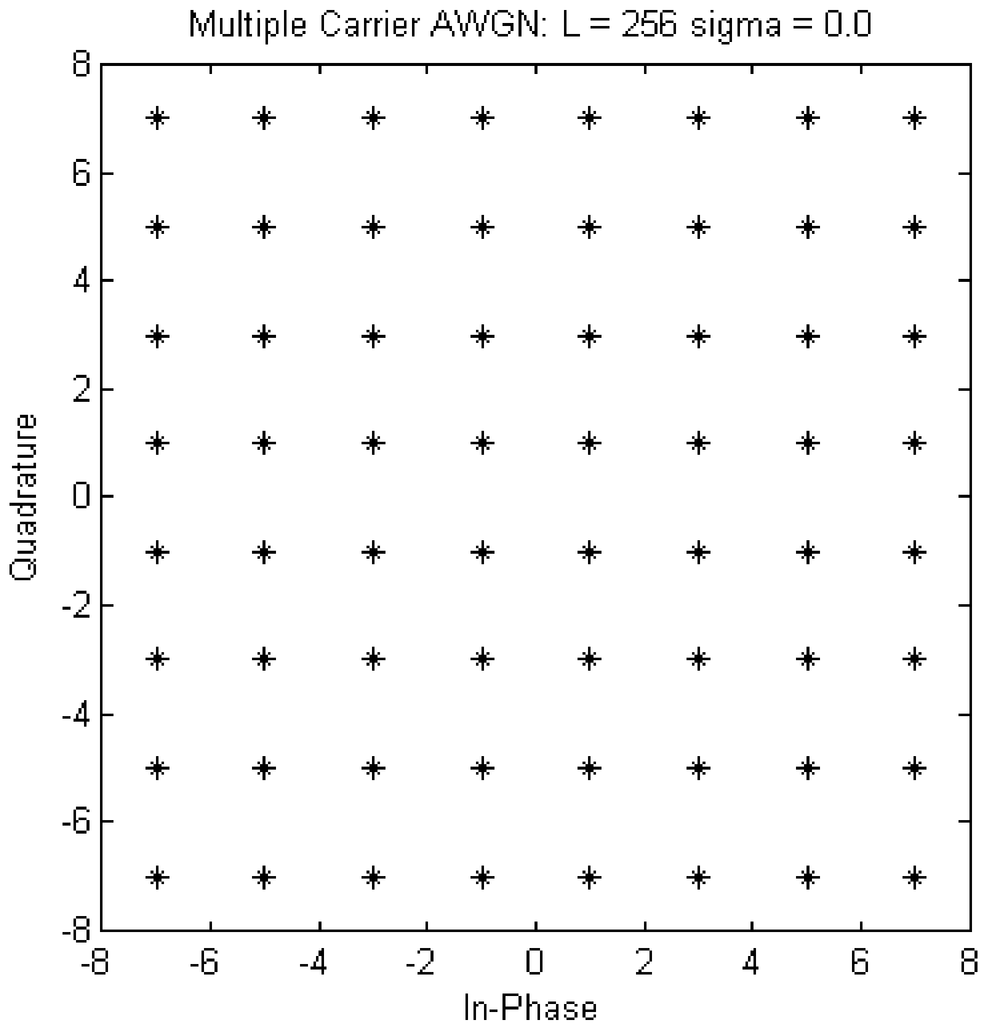

Multiple Carrier AWGN: L = 64 sigma = 0.0
Quadrature
In-Phase
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

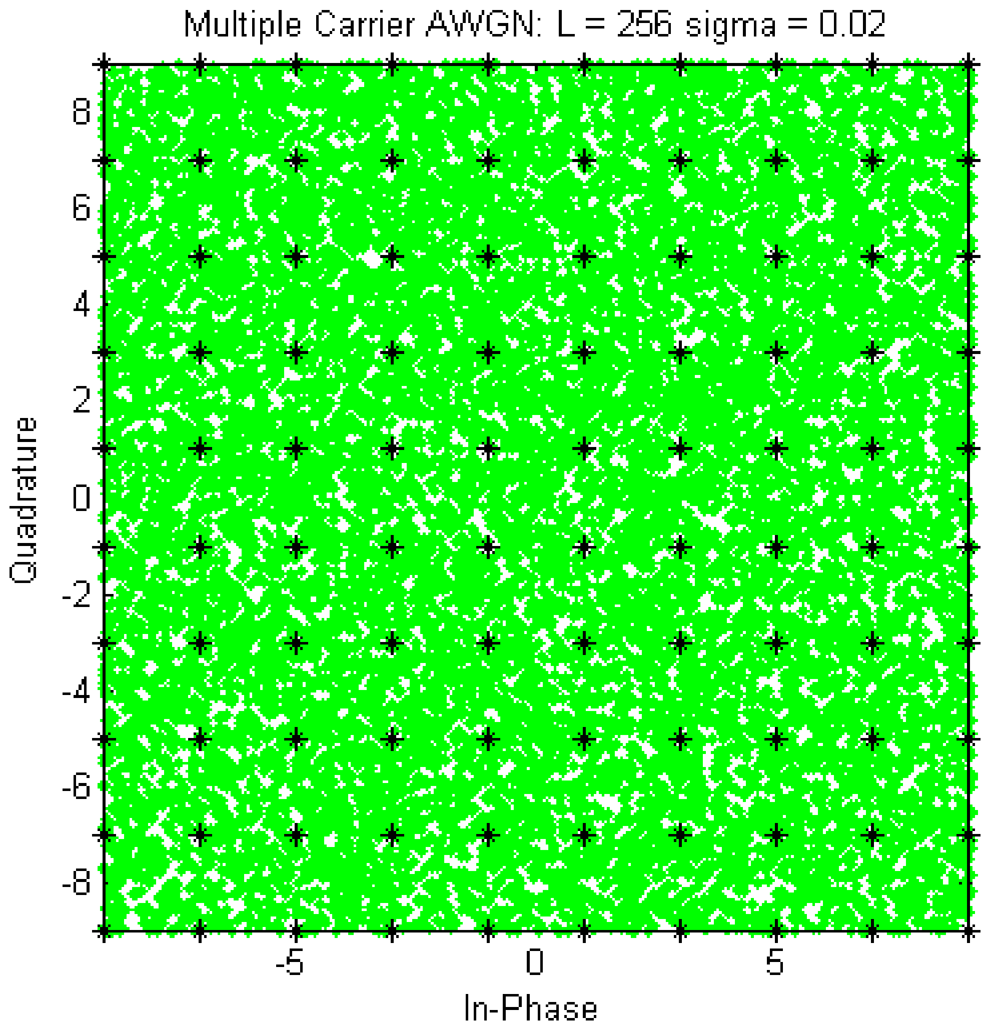

Multiple Carrier AWGN: L = 64 sigma = 0.02
Quadrature
In-Phase
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

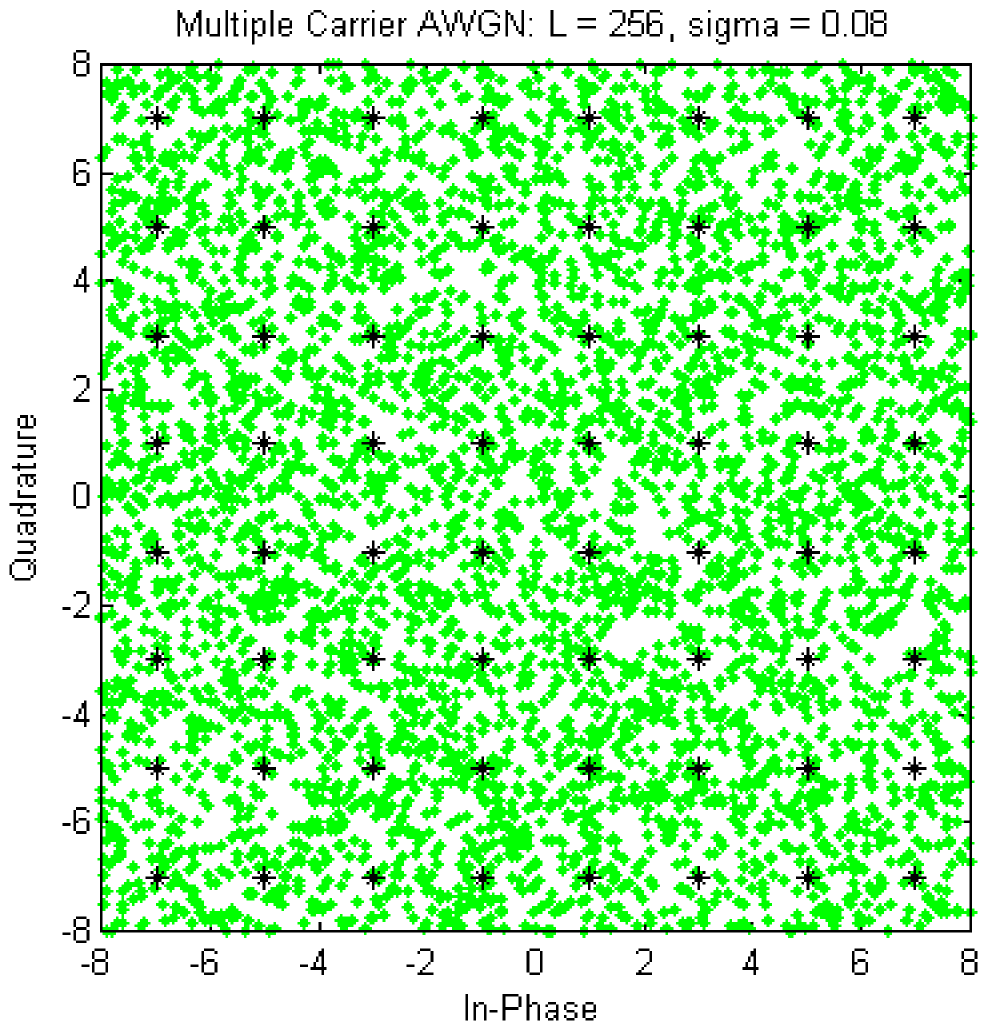

Multiple Carrier AWGN: L = 64, sigma = 0.08
Quadrature
In-Phase
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6


Single Carrier: L = 256
Quadrature
In-Phase
8
6
4
2
0
-2
-4
-6
-8
-8
-6
-4
-2
0
2
4
6
8


Multiple Carrier: L = 256
Quadrature
In-Phase
8
6
4
2
0
-2
-4
-6
-8
-8
-6
-4
-2
0
2
4
6
8


Multiple Carrier AWGN: L = 256 sigma = 0.0
Quadrature
In-Phase
8
6
4
2
0
-2
-4
-6
-8
-8
-6
-4
-2
0
2
4
6
8


Multiple Carrier AWGN: L = 256 sigma = 0.02
Quadrature
In-Phase
8
6
4
2
0
-2
-4
-6
-8
-5
0
5



Clean Up for next run

```
clear all
```

```
end
```

Published with MATLAB® R2014a