# Table of Contents



# loading the image and extracting the signal in 6 bit

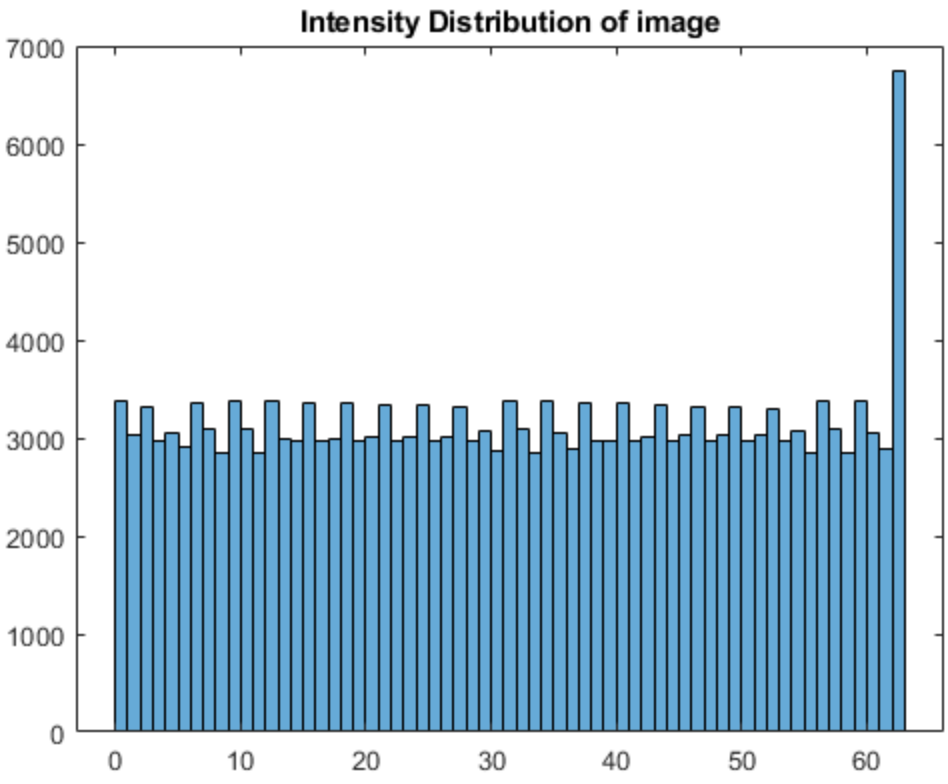
clearing workspace

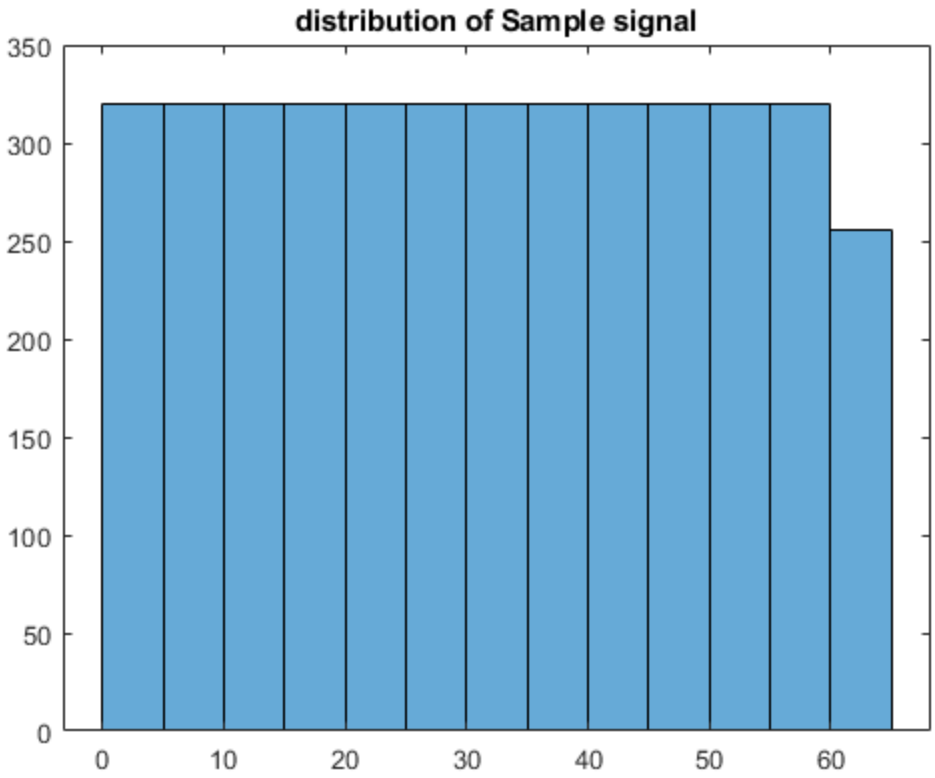
```
clear all;
close all;
clc;
% reading the image
img=imread('grey_scale.jpg');
% converting the image to gray scale
img1=rgb2gray(img);
% initializing img vector to store the image using 64 symbols
img2=zeros(423,470);
for i=1:1:length(img1(:,1))
    for j=1:1:length(img1(1,:))
        temp=double(img1(i,j));
        img2(i,j)=floor(temp/4);
    end
end
%ideal input sample
Sample= zeros(64);
% initializing it with values which are equiprobable
for i=1:64
    Sample(i,:)=i-1;
end
img3=zeros(64);
% converting into double
for i=1:1:length(Sample(:,1))
    for j=1:1:length(Sample(1,:))
        temp=double(Sample(i,j));
        img3(i,j)=temp;
    end
end
```

# Study the distribution of intensities

```
histogram(img2)
title('Intensity Distribution of image');
```

```
figure
histogram(img3)
title('distribution of Sample signal');
```






# Initializing the symbols to be mapped

for the image

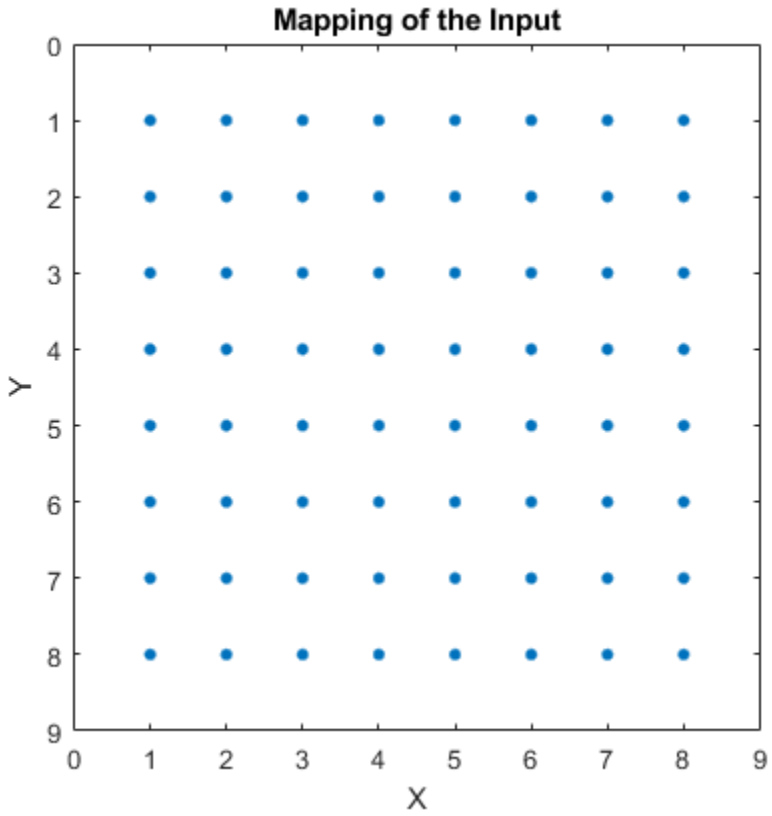
```
 X=zeros(1,8);
 C=zeros(8);
 Y=zeros(1,8);
% for the samples
 X1=zeros(1,8);
 C1=zeros(8);
 Y1=zeros(1,8);
```

computing X,Y,C for the image

```
for i=1:423
    for j=1:470
        temp=img2(i,j);
        X(1,floor(temp/8)+1)=X(1,floor(temp/8)+1)+1;
        Y(1,rem(temp,8)+1)=Y(1,rem(temp,8)+1)+1;
        C(floor(temp/8)+1,rem(temp,8)+1)=
 C(floor(temp/8)+1,rem(temp,8)+1)+1;
    end
end
% Computing X1,Y1,C1 for the Sample signal
for i=1:64
    for j=1:64
```

```
        temp=Sample(i,j);
        X1(1,floor(temp/8)+1)=X1(1,floor(temp/8)+1)+1;
        Y1(1,rem(temp,8)+1)=Y1(1,rem(temp,8)+1)+1;
        C1(floor(temp/8)+1,rem(temp,8)+1)=
 C1(floor(temp/8)+1,rem(temp,8)+1)+1;
    end
end

figure
spy(C)
title('Mapping of the Input')
xlabel('X')
ylabel('Y')
```



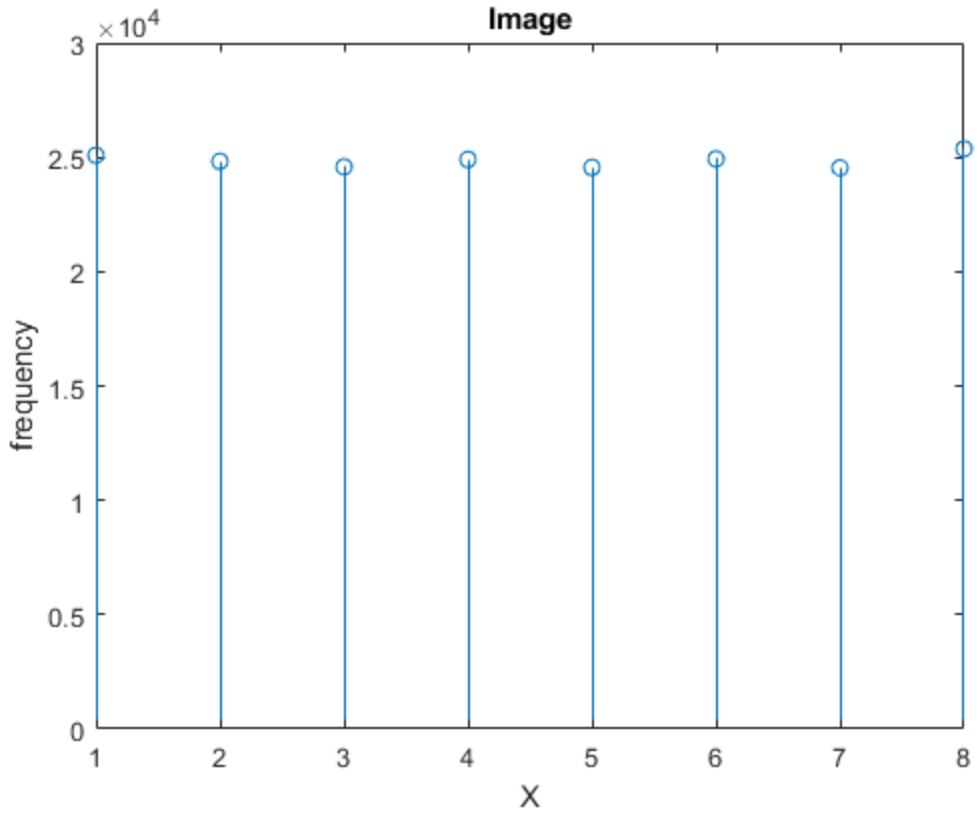

# Studying the distribution of the random variables

```
figure
stem(X)
xlabel('X')
ylabel('frequency')
title('Image')
figure
stem(Y)
xlabel('Y')
```

```
ylabel('frequency')
title('Image')

figure
stem(X1)
xlabel('X1')
ylabel('frequency')
title('Sample Image')
figure
stem(Y1)
xlabel('Y1')
ylabel('frequency')
title('Sample Image')

figure
stem(X)
ylim([min(X),max(X)]);
xlabel('X')
ylabel('frequency')
title('showing the deviation from ideality')
figure
stem(Y)
xlabel('Y')
ylabel('frequency')
ylim([min(Y),max(Y)]);
title('showing the deviation from ideality')
```



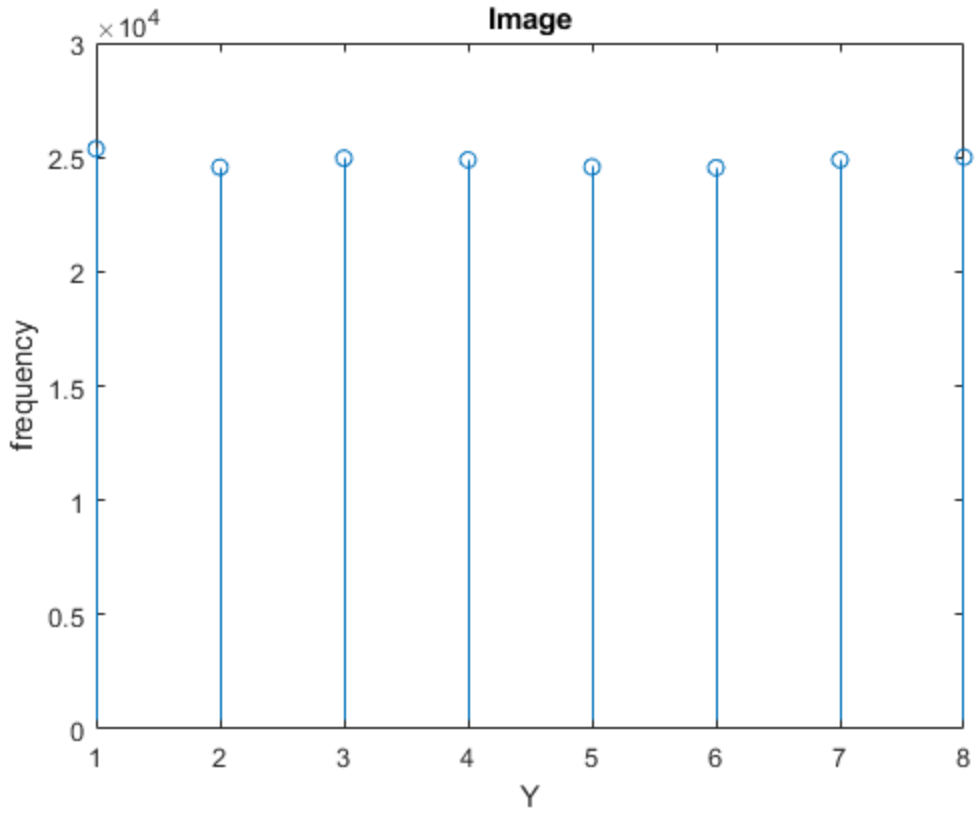
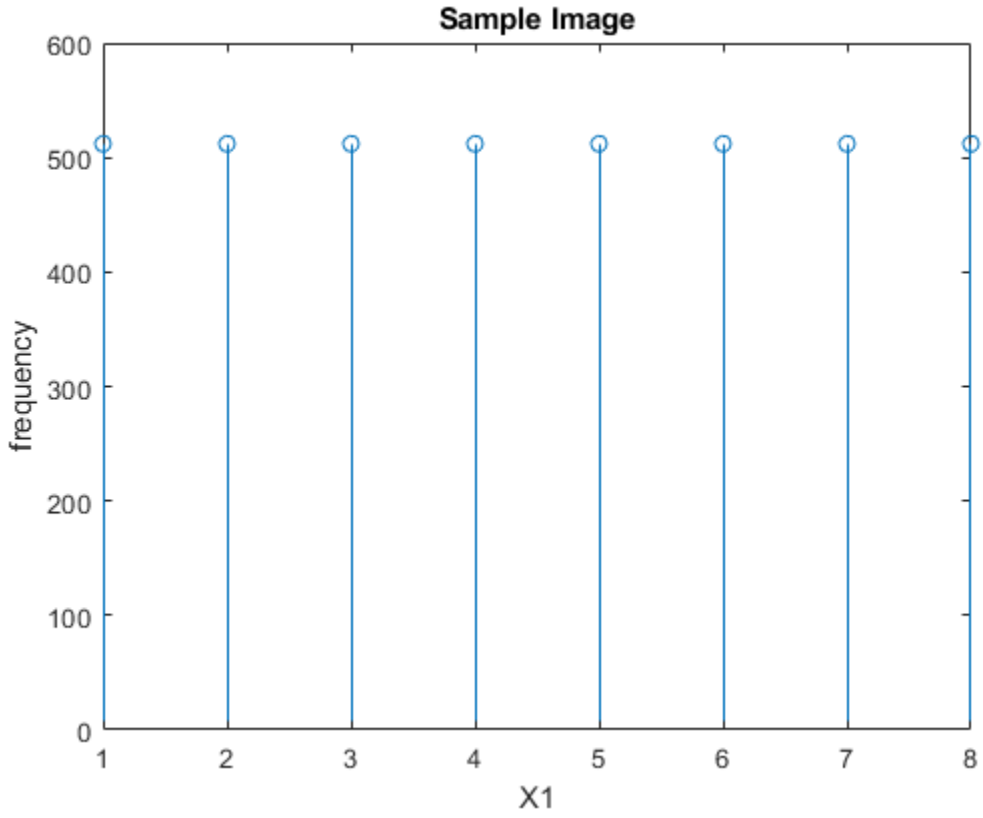
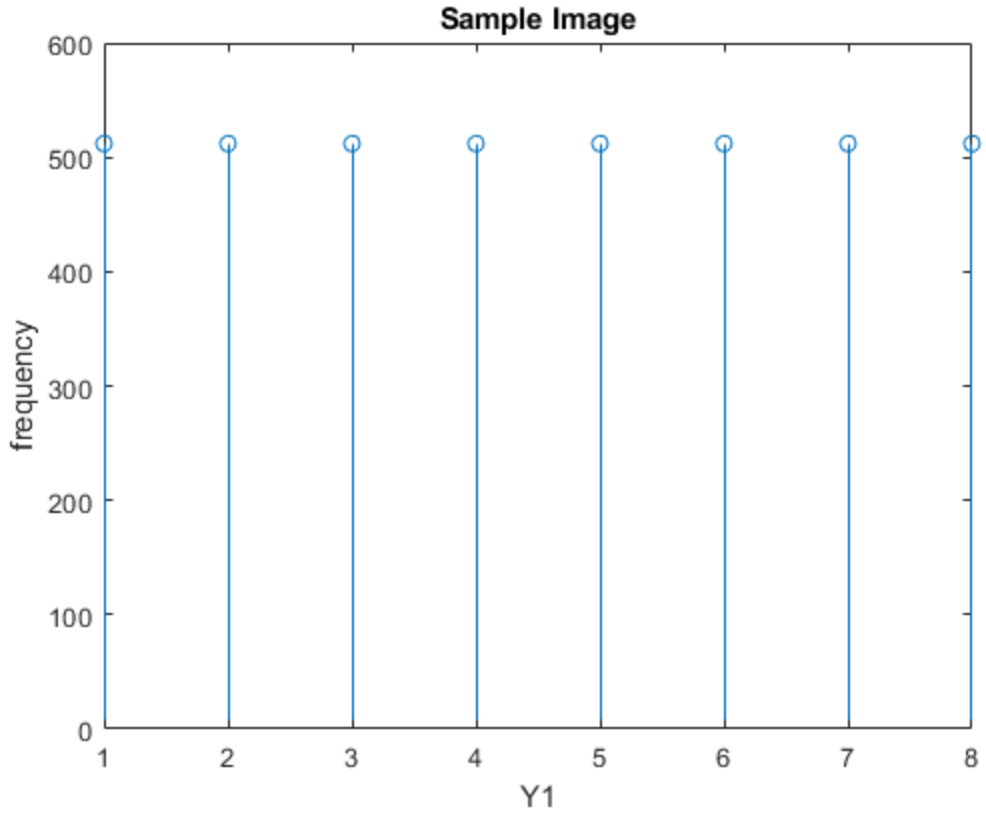
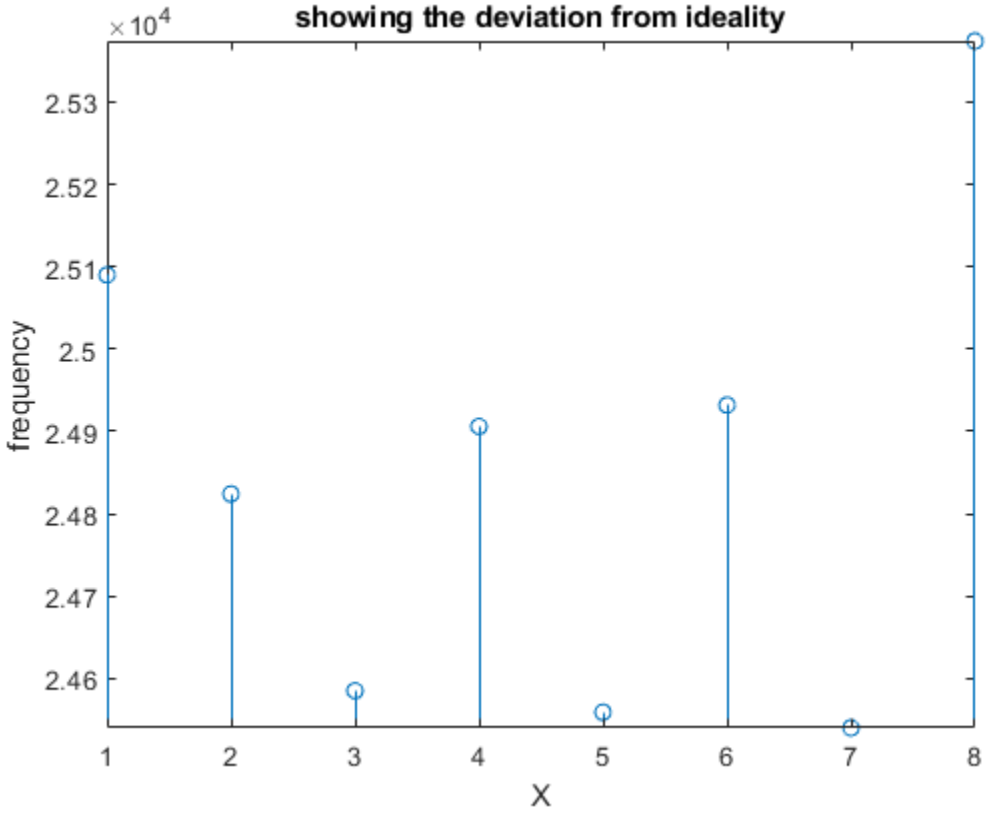
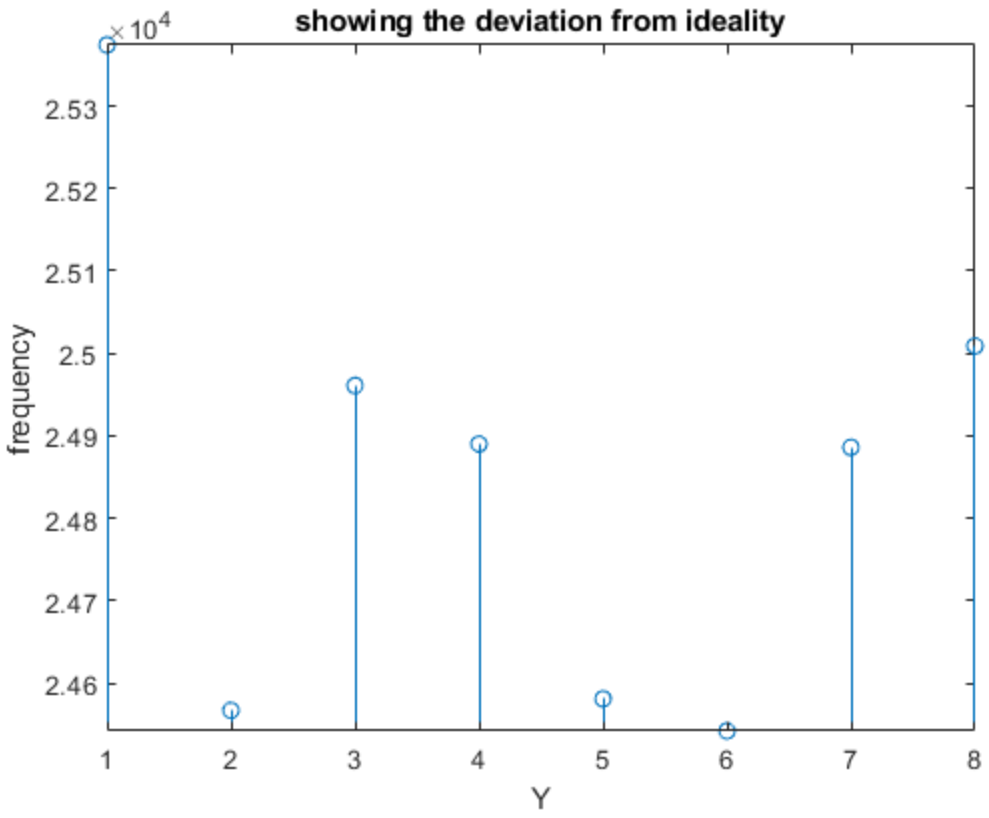


Image
×10^4
3
2.5
2
1.5
1
0.5
0
frequency
1
2
3
4
5
6
7
8
Y


Sample Image
600
500
400
300
200
100
0
frequency
1
2
3
4
5
6
7
8
X1


Sample Image
600
500
400
300
200
100
0
frequency
1
2
3
4
5
6
7
8
Y1


showing the deviation from ideality
×10^4
2.53
2.52
2.51
2.5
2.49
2.48
2.47
2.46
frequency
1
2
3
4
5
6
7
8
X




# Applying Box Muller Theorem

Initialising the values

```
J=zeros(8);
J2=zeros(8);
for i=1:8
    for j=1:8
        temp1=sqrt(-2*log((i/8)-0.12))*cos(2*pi*(j/8)-0.12);
        temp1=temp1*C(i,j);
        temp2=sqrt(-2*log((i/8)-0.12))*(sin(2*pi*(j/8)-.12));
        temp2=temp2*C(i,j);

        J(i,j)=temp1;
        J2(i,j)=temp2;
    end
end

I=zeros(8);
I2=zeros(8);
for i=1:8
    for j=1:8
        temp1=sqrt(-2*log((i/8)-0.12))*cos(2*pi*(j/8-0.12));
       temp1=temp1*C1(i,j);
        temp2=sqrt(-2*log(i/8-0.12))*(sin(2*pi*(j/8-0.12)));
       temp2=temp2*C1(i,j);
```

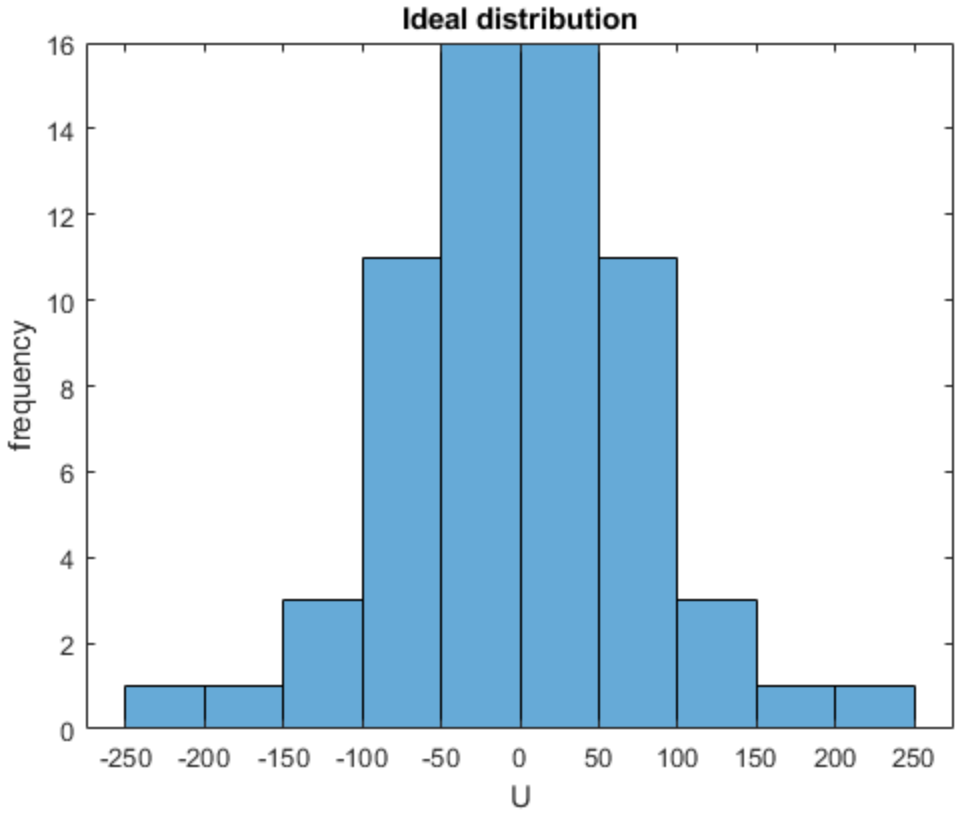
```
        I(i,j)=temp1;
        I2(i,j)=temp2;
    end
end
```

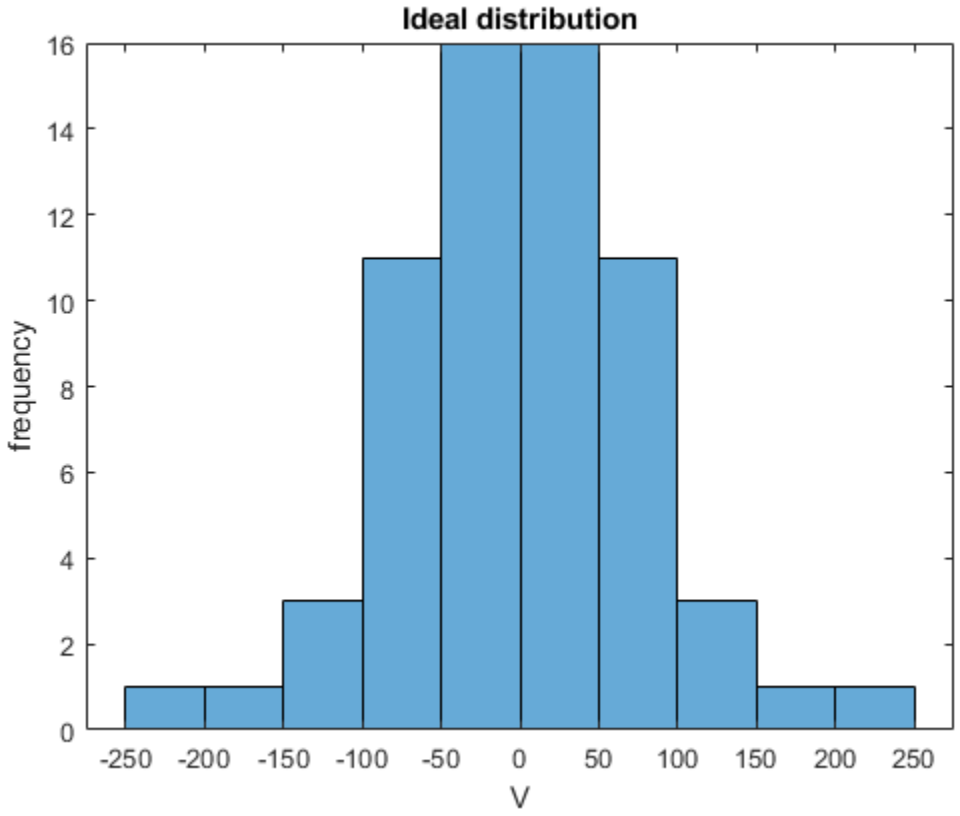
# Plotting and comparing the results.

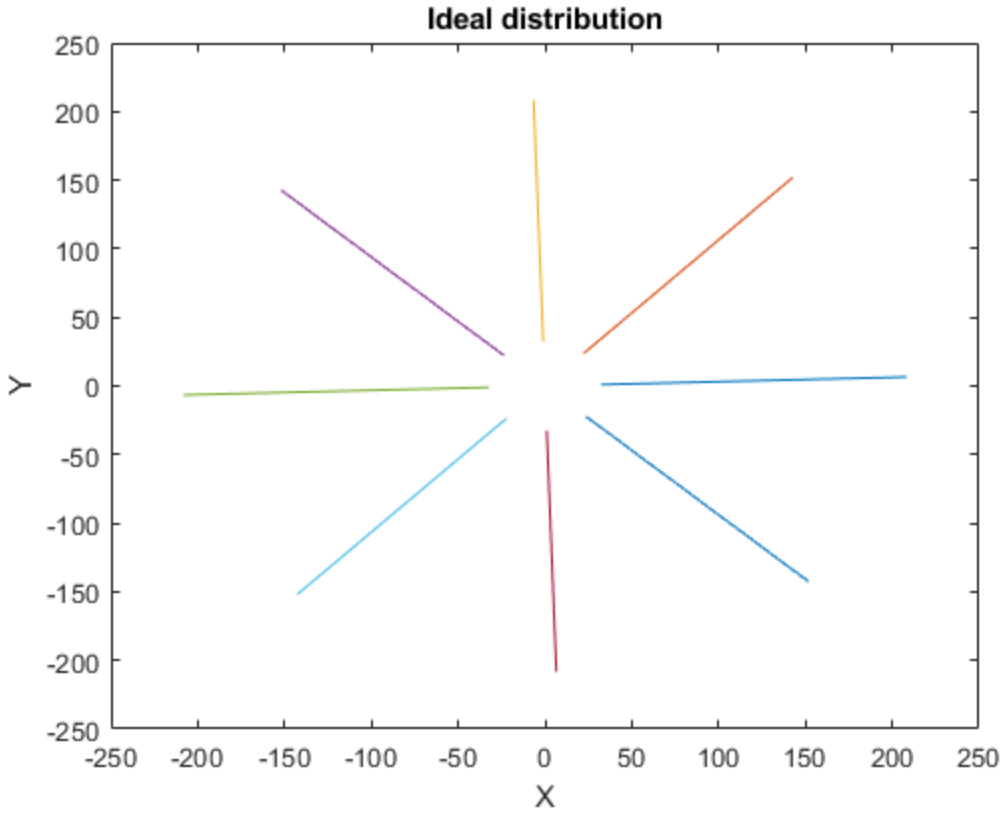
```
figure
histogram(I)
title('Ideal distribution')
xlabel('U')
ylabel('frequency')

figure
histogram(I2)
title('Ideal distribution')
xlabel('V')
ylabel('frequency')
figure
plot(I,I2)
title('Ideal distribution')
xlabel('X')
ylabel('Y')


figure
histogram(J)
title('image distribution')
xlabel('U')
ylabel('frequncy')
figure
histogram(J2)
title('Image distribution')
xlabel('V')
ylabel('frequncy')
figure
plot(J,J2)
title('Image distribution')
xlabel('U')
ylabel('V')
```


Ideal distribution
frequency
U
0
2
4
6
8
10
12
14
16
-250
-200
-150
-100
-50
0
50
100
150
200
250


Ideal distribution
frequency
V
0
2
4
6
8
10
12
14
16
-250
-200
-150
-100
-50
0
50
100
150
200
250


Ideal distribution
250
200
150
100
50
0
-50
-100
-150
-200
-250
Y
-250 -200 -150 -100 -50 0 50 100 150 200 250
X

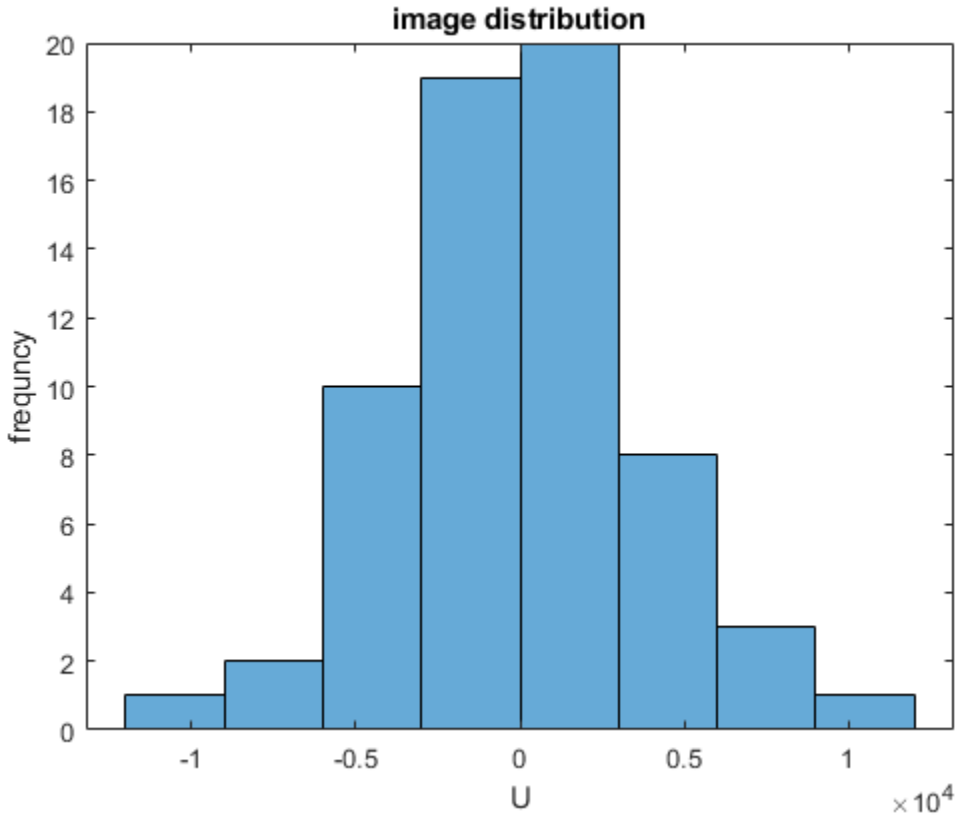


image distribution
20
18
16
14
12
10
8
6
4
2
0
frequncy
-1 -0.5 0 0.5 1
U
×10^4

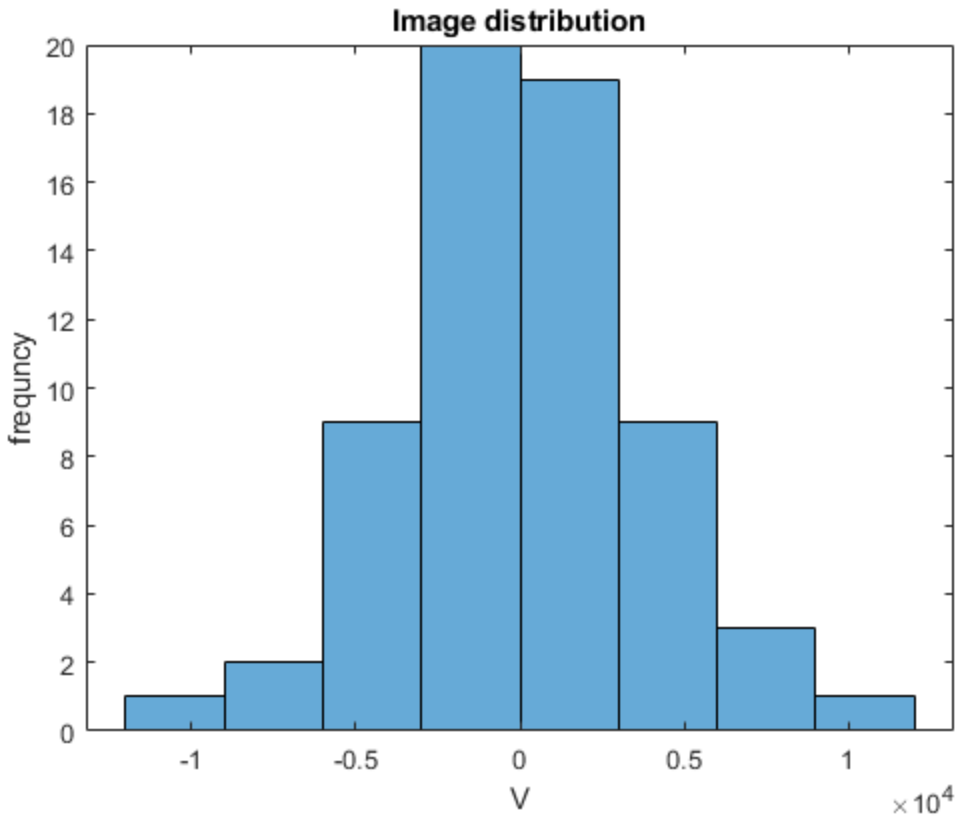


Image distribution
frequncy
V
×10^4

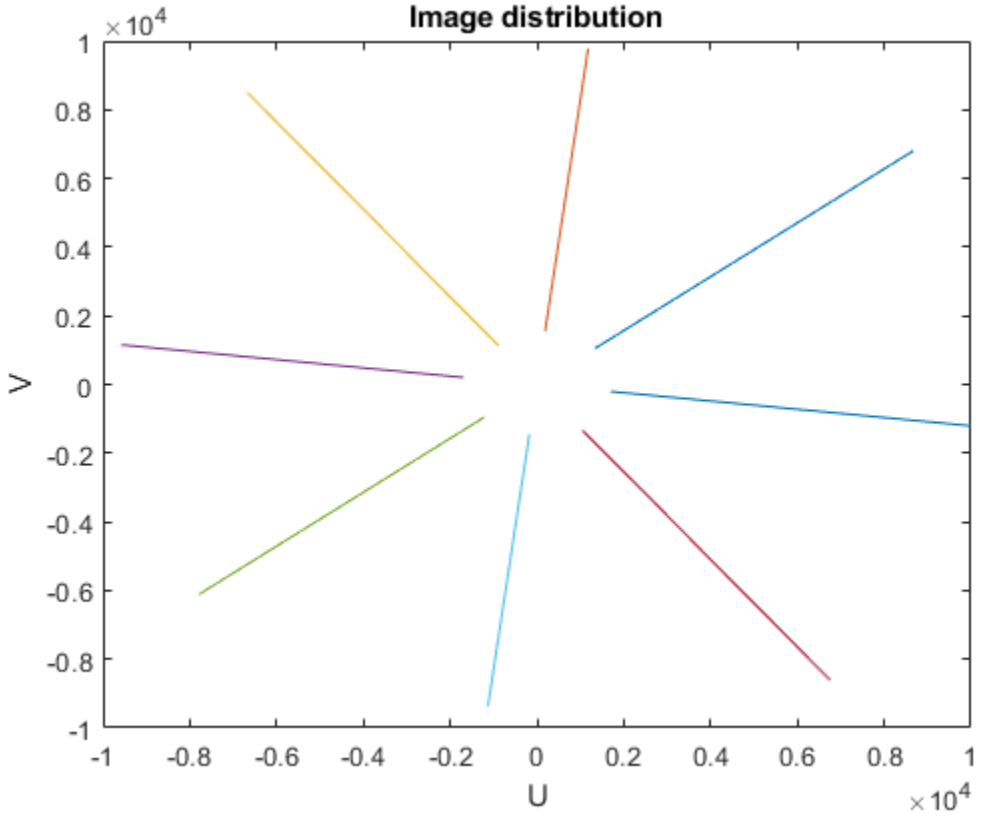


Image distribution
×10^4
V
U
×10^4

*Published with MATLAB® R2017b*