EED 350-SP2021: DIGITAL COMMUNICATION PROJECT REPORT

NAME: ASHNA JAIN
ROLL NO. : 1810110046
SUBMITTED TO: PROFESSOR VIJAY KUMAR CHAKKA

TITLE : Approaching the Gaussian Channel Capacity with APSK Constellations

- ***Abstract:***

We consider the Gaussian channel with power constraint P.A gap exists between the channel capacity and the highest achievable rate of equiprobable uniformly spaced signal because of the power constraint and limited bandwidth.In this letter, focus is on constellation shaping using Box Muller theorem. We give a construction of amplitude and phase-shift keying (APSK) constellations with equiprobable signaling that achieve the Gaussian capacity as the number of constellation points goes to infinity.

- ***Introduction:***

One of the most frequently encountered and practically relevant communication channel mode is AWGN (Additive White Gaussian Noise) for both analog and digital communication. AWGN can be represented as : y(t) = x(t) + n(t) , where x(t) = Transmitted signal , n(t)= Additive Gaussian Noise channel (0 mean and normal distribution with variance N_0) and y(t)= Received signal

The Gaussian channel capacity is defined by Shannon and Hartley Theorem . The expression for the channel capacity is :

$$\mathbf{C = \frac{1}{2}\log_2(1 + SNR)} \text{ bits/dimension ,}$$

where signal-to-noise ratio (SNR) = P/N_0 , P is the average power constraint of the input signal and N_0 the noise variance.

In digital communications systems, approaching the Gaussian capacity is a challenging problem. Indeed the optimal input distribution is continuous while in practice the signal is chosen from a finite constellation of points.

Hence the final aim is to achieve maximum capacity of channel : Subject to power constraint. The main task is the construction of APSK modulations with equiprobable signaling that achieve the Gaussian capacity as the number of constellation points goes to infinity. To that end, we rely on the Box-Muller theorem .

The research results may benefit some practical systems as the digital video broadcasting (DVB) standards that implement APSK constellations with large cardinality, up to 256 .

- ***Proposed Method in Research Paper for Approaching Gaussian Capacity:***

Consider two discrete random variables that converge weakly to the uniform distribution over (0,1). Applying a *Box-Muller transform* to that pairs of random variables will generate a set of two dimensional points with a Gaussian shape, the key ingredient to approach the Gaussian capacity.

Box Muller Theorem states that - Let U and V be two independent random variables that are uniformly distributed in the interval (0, 1). As a result two independent random variables X and Y with standard normal distribution is given by :

$$X = \sqrt{-2\log_e U}\cos(2\pi V)$$

$$Y = \sqrt{-2\log_e U}\sin(2\pi V)$$

Consequently we introduce two sequences $\mathbf{U_n}$ and $\mathbf{V_n}$: two discrete random variables uniformly distributed. These two sequences consists of the equal number of elements between 0 and 1. These elements are uniformly distributed on discrete set $\mathbf{S_n}$ **.** The set is defined as :

$$\mathcal{S}_n = \left\{ \frac{1}{2n} + \frac{k}{n} \mid k \in \mathbb{N}_n \right\}$$

Let n ≥ 1 and N_n is set of integers from 0 to (n -1) .

The expression verifies that $|S_n| = n$ and $S_n \subset (0, 1)$.

Based on the Box –Muller transform a shaping function is defined $\boldsymbol{\psi} : (0,1)^2 \rightarrow R^2$
It is defined represented by :

$$\varphi(x,y) = \left(\sqrt{-P\log_e x}\cos(2\pi y), \sqrt{-P\log_e x}\sin(2\pi y)\right)$$

Here , P is the power constraint of the input signal. Comparing to X and Y random variable equation obtained in ***Box Muller Transform***, there is a scaling factor of $\sqrt{\frac{P}{2}}$ as the optimal input distribution is Gaussian with variance P/2 per dimension. The shaping function objective is to give the Gaussian shape to our constellation.

Finally, we introduce the random vector $W_n = \boldsymbol{\psi}(U_n,V_n)$. By construction, W_n is a random vector uniformly distributed on a **APSK Constellation set $\mathbf{C_n}$** of n^2 (n x n) points in R^2. The points in C_n exhibit a Gaussian shape and are distributed on n circles, each circle containing n points.

The set S_n ensures that U_n is never equal to zero, avoiding problem with the logarithm, and also that the constellation points are uniformly distributed on each circle.

➢ *Implementation of the Approach using MATLAB:*

PART-A: USING NON-UNIFORM DISTRIBUTED INTENSITY IMAGE

STEP 1- GENERATION OF APSK CONSTELLATION – C_n

1. A sample RGB image of 400 x 600 size is taken as input and converted to Grey Scale for further simplification.

2. The grey scale image is changed from intensity range (0,255) to (0,63) in order to generate 64 APSK . Hence a 6 bit signal is extracted.

3. The intensity distribution is plotted. In this case it is non-uniform as shown in figure 1. Hence it has to be approximated as a uniform distribution to make it equiprobable signal. In order to make the intensity distribution histogram uniform , *histeq()* function can be used available in MATLAB. However due to some error this function was not working in the code. Hence the same non – uniform distribution image has been used here. Since the given approach only works on the equiprobable signal (uniformly distributed signal), the same experiment is performed later on an ideal sample image with an uniform distribution of intensity for the correct verification of results given in the research paper.

4. These values are then mapped to general variables taken as U , V and so ideally they should also follow uniform distribution. In this case it would be non-uniform (1 x 8) and V (1 x 8) vector acting as the x and y coordinates respectively of the constellation. These are are mapped to constellation C of size 8 x 8. Plots are shown in figure 2.

5. Hence a 64 APSK (n = 8): from a set of equiprobable and uniformly distributed points in R^2, the Constellation is obtained as shown in figure 3 .

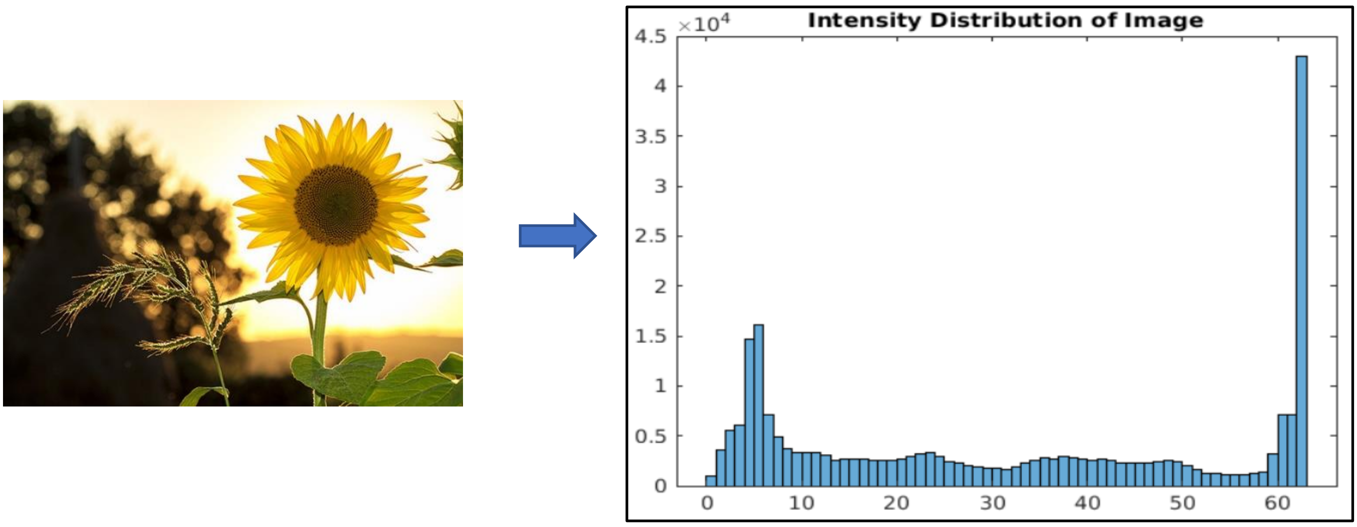

Figure 1 : Histogram Plot of Non-Uniform Intensity Distribution of the given RGB image

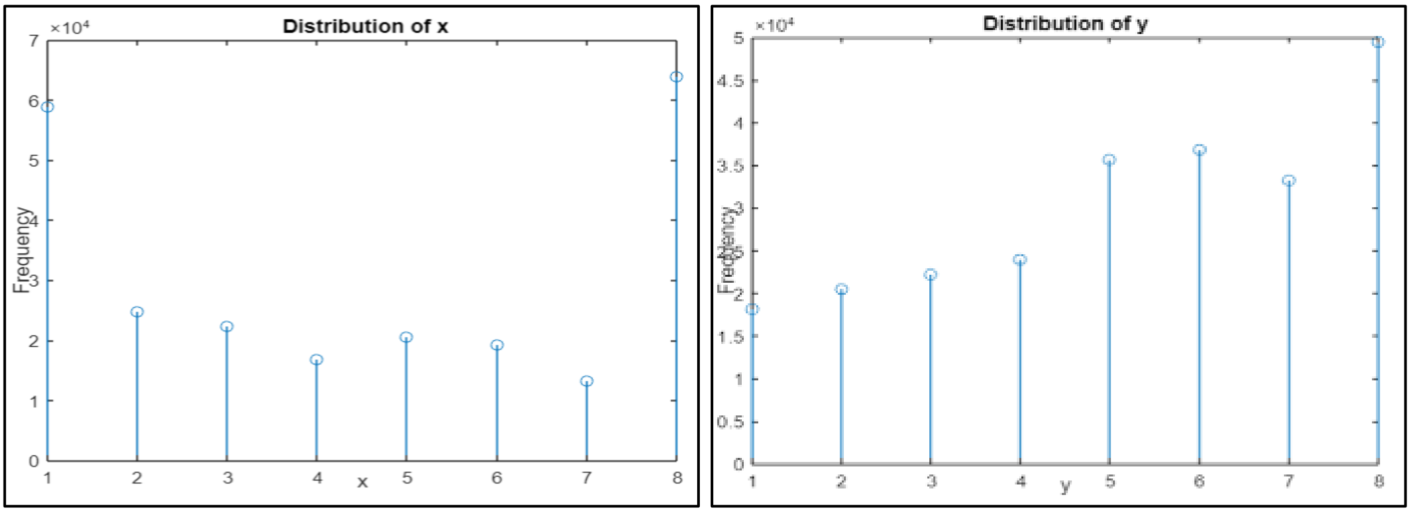

Figure 2 : Distribution of Random Variables-Non- Uniform Frequency

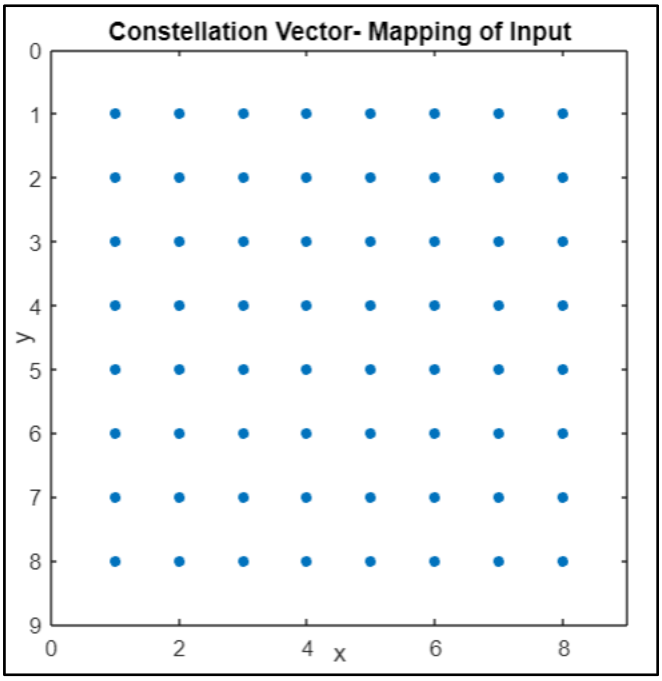

Figure 3 :64 APSK Constellation (8 x 8) formed after mapping U, V using spy() function in MATLAB.

CODE -FOR APSK CONSTELLATION GENERATION:

```
clc;
close all;
% Read the image
image=imread('sunflower.tif');
figure
imshow(image)
% Converting the image to grey scale so that it can be represented with 64
% Symbols(intensities) , where n=6 (number of bits)
image1=rgb2gray(image);
image2=zeros(400,600);

for i=1:1:length(image1(:,1))
    for j=1:1:length(image1(1,:))
        temp=double(image1(i,j));
        image2(i,j)=floor(temp/4);
    end
end
%% Intensity analysis
% shows the probability of various intensities
%J=histeq(img2);
%imhist(J)
histogram(img2)
title('Intensity Distribution of Image')
```

Input Image Loading and conversion to grey scale. Later 6 bit signal extracted

Converting to double

eqhist() function commented as it not work in the code

```
%% Initiating U,V,C and the intensities are mapped to these symbols
%% C - Constellation Vector , U= Row Vector of C (x) , V= Column Vector of C (y)
 U=zeros(1,8);
 C=zeros(8,8);
 V=zeros(1,8);
 % computing the frequency of occurance of each symbols for easy processing
 for i=1:400
    for j=1:600
        temp=image2(i,j);
        U(1,floor(temp/8)+1)=U(1,floor(temp/8)+1)+1;
        V(1,rem(temp,8)+1)=V(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
%% Visualising the distribution the variables follow.
figure
spy(C)
xlabel('x');
ylabel('y');
title('Constellation Vector- Mapping of Input')
figure
stem(U)
xlabel('x')
ylabel('Frequency')
title(' Distribution of x')
figure
stem(V)
xlabel('y')
ylabel('Frequency')
title('Distribution of y')
```

Mapping of U and V variables to 8 x 8 APSK constellation matrix

Plotting of all the graphs .

The grey scale matrix of the image was changed from uint8 to double matrix so that any mathematical operation could be performed correctly.

The above histograms and stem graphs has been plotted using MATLAB as shown in the above code.

STEP 2- APPLYING BOX-MULLER TRANSFORM:

1. The shaping function is applied on the obtained APSK Constellation using the Box Muller transform by one to one mapping.
2. The Random Variables U and V consisting of 8 values uniformly distributed in range (0,1) are generated using nested loop. In this case same values for U and V are taken.
3. Gaussian Variables X and Y obtained after the Box-Muller transform is applied and plotted .
4. Since the image intensity distribution was not uniform , the shaping function has not generated a proper gaussian- shaped constellation.
5. As a result ,the circles are not exactly equal to n and each circle has different number of points as shown in figure 4.

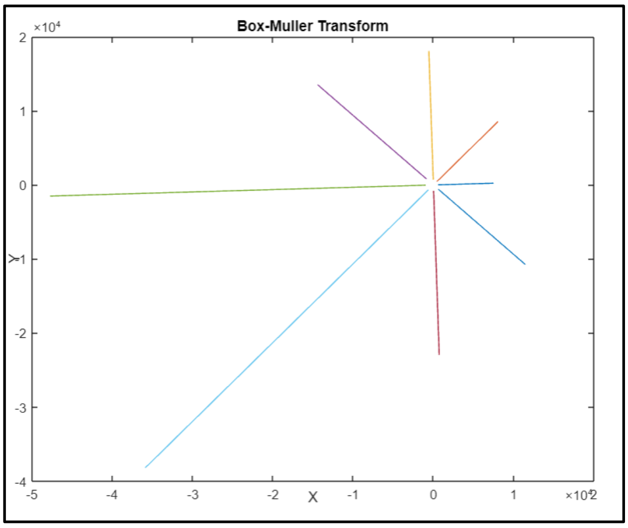

Figure 4 :BOX MULLER Transform of Input Image

The distribution of X and Y Random Variable is given below in figure 5. The distribution is imperfect and discontinuous Gaussian . Consequently the results were not same as expected.

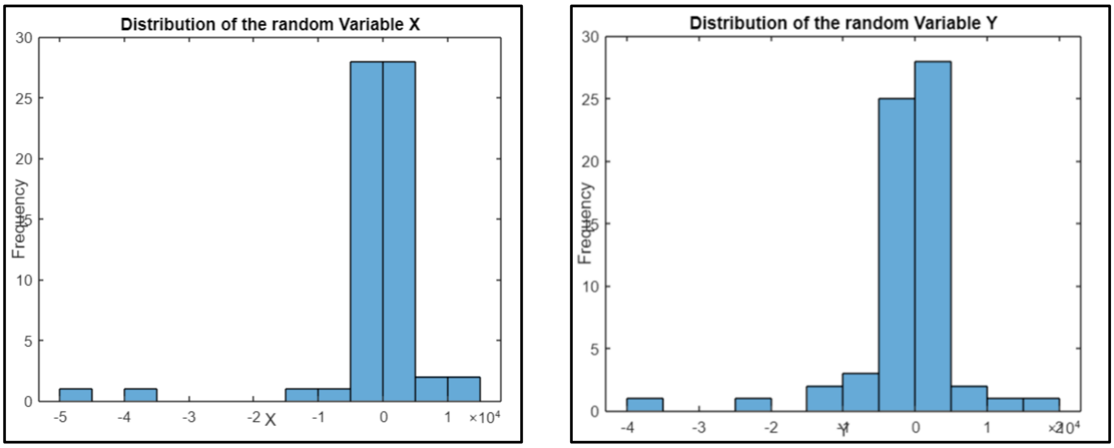

Figure 5:Histogram plot of the distribution of X and Y in MATLAB

CODE-FOR BOX MULLER TRANSFORM :

```
%% Applying Box-Muller transform
% U and V are transformed into X and Y gaussian symbols
% initialising X and Y
% Generating U , V random variables between (0,1) by in for Loop
X=zeros(8);
Y=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/8-0.12));
        temp2=temp2*C(i,j);
        X(i,j)=temp1;
        Y(i,j)=temp2;
    end
end
```

```
%% Analysis of the random variable X and Y after the Box muller transform
figure
histogram(X)
title('Distribution of the random Variable X')
xlabel('X')
ylabel('Frequency')
figure
histogram(Y)
title('Distribution of the random Variable Y')
xlabel('Y')
ylabel('Frequency')
figure
plot(X,Y)
title('Box-Muller Transform')
xlabel('X')
ylabel('Y')
```

The random variable sequences U_n and V_n ,mentioned in the Box Muller Theorem has been made by using the nested for loop as shown in the code. Here 8 elements have been calculated for each sequence between 0 and 1 at a uniform distribution. In this case the element values of both random variable sequences are considered equal.

PART-B: USING UNIFORM DISTRIBUTED INTENSITY SAMPLE IMAGE

Generation of Ideal Signal:

- Practically it was difficult to find an image with uniform distribution. Hence an ideal sample is created with equiprobable distribution to verify the results. The ideal sample image is a matrix of 64 x 64 values with each column having values from 0 to 63.
- Since the ideal image intensity distribution is uniform, the frequency of each mapped point is also equal. The results are shown in figure 6 and figure 7.

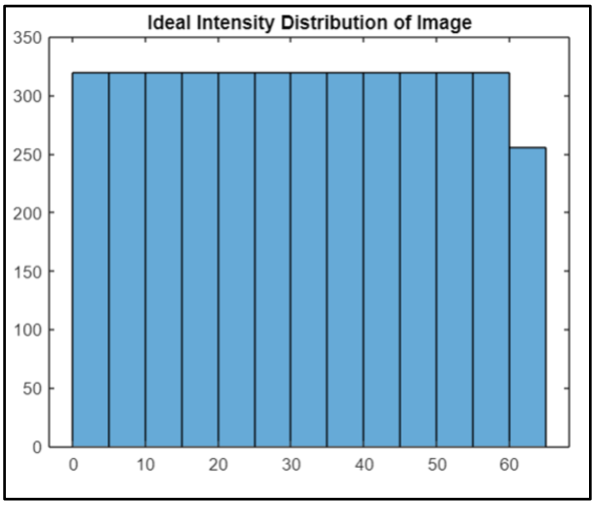

Figure 6: Equiprobable distribution of Ideal sample

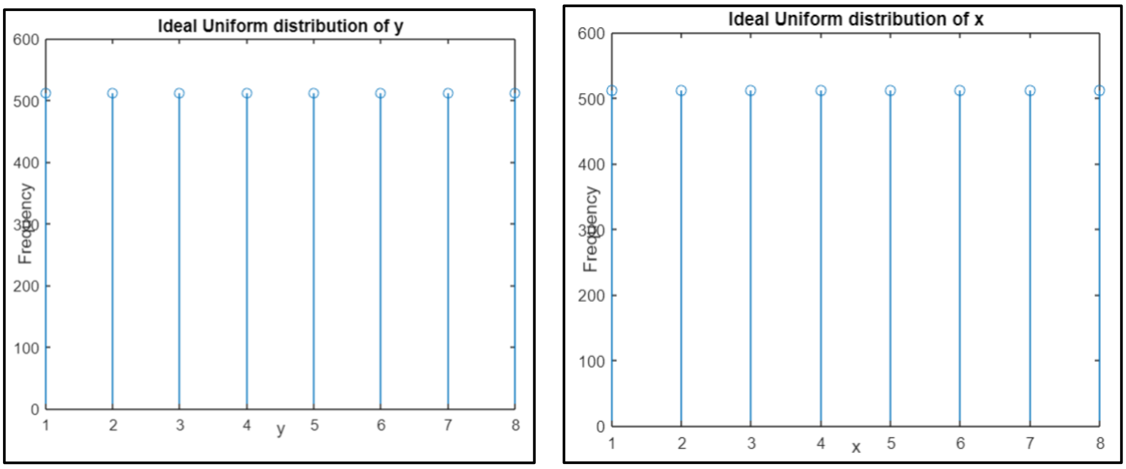

Figure 7: Equiprobable distribution of x and y frequency

Applying Box-Muller on Ideal Sample

Since the distribution was uniform , the shaping function generates a gaussian constellation distributed on 8 circles and each circle containing 8 points as shown in figure 8. Hence it is verified that the results shown in the research paper are correct .

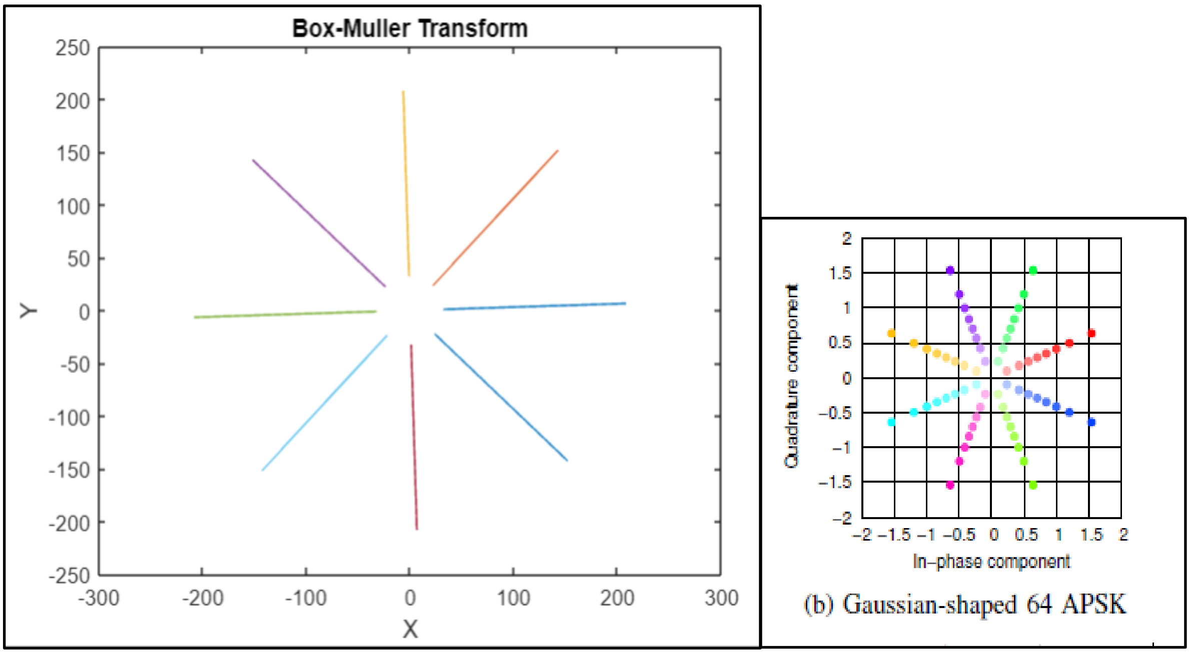

Figure 8: Box Muller Transform of Ideal Sample Image

The histogram plot of X and Y random variable has a perfect gaussian distribution as shown in the figure 9.The gaussian distribution of the APSK constellation has been achieved using the non-linear Box Muller Transform. Further mathematical conditions are given which proves that these gaussian constellation can achieve channel capacity if the number of points (n) in the constellation tends to infinity. These conditions are discussed in the next section.

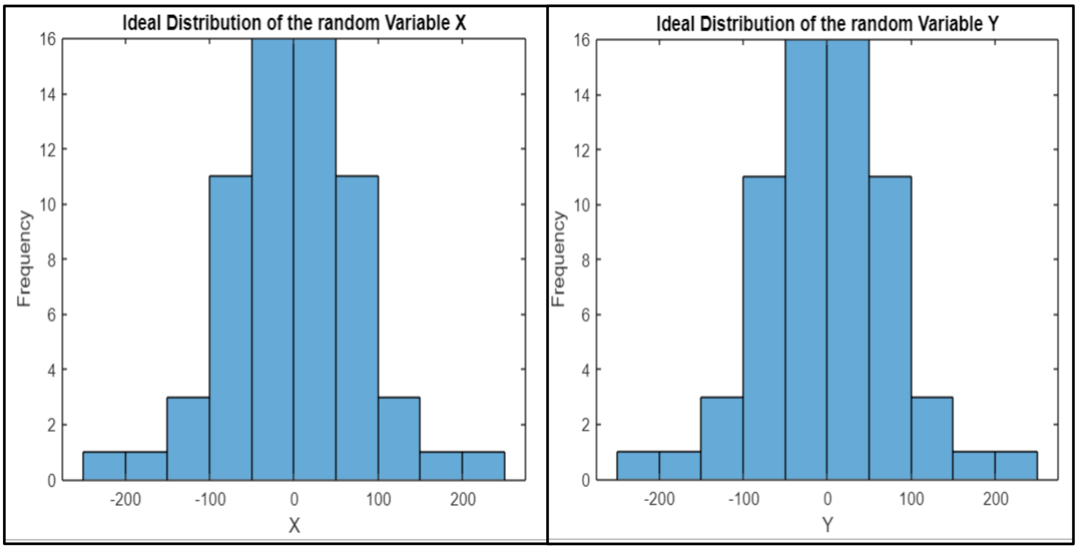

Figure 9: Histogram Plot of Equiprobable distribution of X and Y in MATLAB

CODE- FOR GENERATION OF IDEAL SAMPLE

```
clc;
close all;
% Ideal Sample Input
sample=zeros(64,64);
%% Initializing 64 symbol equiprobable values
for i=1:64
    sample(i,:)=i-1;
end

img=zeros(64);
%converting to double
for i=1:1:length(sample(:,1))
    for j=1:1:length(sample(1,:))
        temp=double(sample(i,j));
        img(i,j)=floor(temp);
    end
end
```

- ***Results and Conclusion:***

The following theorem is the main result of the paper:

Theorem : The APSK constellations previously designed achieve the Gaussian channel capacity as $n \to \infty$. More formally,

$$\mathbf{I(W_n;W_n + N)} \underset{n\to\infty}{\Longrightarrow} \mathbf{\log_2(1 + SNR)}$$

where $N \sim N(0, N_0)$ is the AWGN with variance N_0 ($N_0/2$ on each dimension).

Followings conditions are taken into consideration which proves the result obtained in the paper:

CONDITION-1

Constellation C_n satisfies the power constraint for n >= 1 .

A lemma is presented that will be helpful to prove that Cn satisfies the average power constraint.

Lemma:

For any integer $k \geqslant 1$,

$$k \log_e k - k \leqslant \sum_{j=0}^{k-1} \log_e \left(j + \frac{1}{2} \right)$$

Therefore the input signal energy is:

$$\begin{aligned}
\mathbb{E}\left[W_n^2\right] &= \frac{1}{n^2} \sum_{\mathbf{w} \in \mathcal{C}_n} \|\mathbf{w}\|^2 \\
&= \frac{1}{n} \sum_{k=0}^{n-1} -P \log_e \left(\frac{1}{2n} + \frac{k}{n} \right) \\
&= -\frac{P}{n} \left(-n \log_e n + \sum_{k=0}^{n-1} \log_e \left(k + \frac{1}{2} \right) \right) \\
&\leqslant P,
\end{aligned}$$

where the last inequality follows from the previous lemma. Thus the constellation Cn verifies the power constraint.

CONDITION-2

The distribution of $W_n = \boldsymbol{\psi}\ (U_n, V_n)$ converges (weakly) to the distribution of a Gaussian variable, denoted W^*, with variance P/2 on each dimension. The mathematical proof this condition is little complex. Therefore the final result is stated by applying Lesbegue's Dominated convergence theorem we obtain:

$$\Phi_{W_n}(\mathrm{t}) \underset{n \to \infty}{\longrightarrow} \Phi_{W^*}(\mathrm{t}).$$

Finally, it follows from the continuity theorem that W_n converges weakly to W^*

CONCLUSION

- The constellation uniformly distributed over C_n satisfies both conditions .We conclude that the constellations approach the Gaussian channel capacity with power constraint P as n $\rightarrow \infty$.
- The APSK design enables to achieve the Gaussian capacity as the constellation size grows to infinity. APSK modulations are better than regular or conventional QAM for example 16 QAM because of the lower number of possible amplitude levels. APSK combines both amplitude-shift keying (ASK) and phase-shift keying (PSK) to increase the symbol-set.
- Although this method has achieved the channel capacity, the convergence speed remains an open question. There are different methods which have been researched where we don't want require infinite constellation points. Moreover the DVB-S2X design reduces the peak-to-average power ratio, a suitable property for practical systems.