**CODE REPORT**

**Theory:**

We’ve fading channel coefficient is,

where, is career frequency

is delay in path.

is attenuation factor

It can be written as,

where, ,

Where x and y are sum of a large set of random variables because of random attenuation () and random delay (. When we add a large set of random component variables, it results a Gaussian distribution by central limit theorem.

Now and assumed to be Gaussian Random variable with probability distribution function are and respectively.

Writing as magnitude and phase form we get,

where, ,

where, the marginal distribution of we get after integration is ( Rayleigh distribution.)

In case of we get the marginal distribution (uniform distribution).

**Code Workflow:**

Using the function

numpy.random.randn(N)

We can generate an array of normally distributed random numbers of size N with mean zero and unity standard variation.

The snippet below

h = (1/numpy.sqrt(2))\*(numpy. random.randn(N) + numpy. random.randn(N)\*1j)

creates an array of complex numbers with random components of size N=100000 with mean zero and variance 1/2.

With the help of abs function we calculated magnitude(modulus) of all elements and stored in array a.

And created the histogram class of above array a with uniform class size of 0.05 using the following snippet

pdfa, bin\_edges = np.histogram(a,np.arange(0,4.05,0.05))

Probability of each histogram class can be calculated as,

And, approximate density of each histogram class can be seen as,

Now plot of amplitude(magnitude) probability density is figured using following snippet,

#get plot of modulus graph

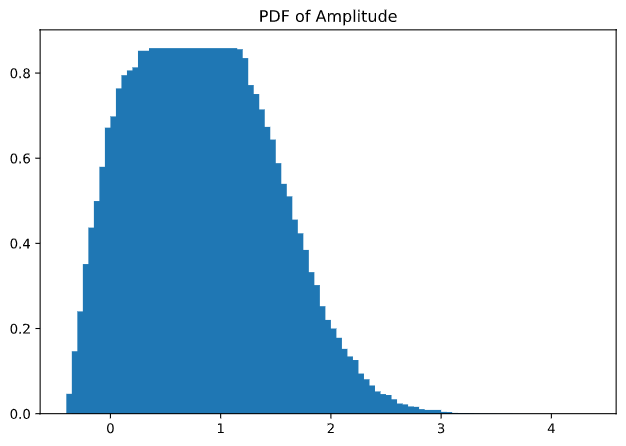
fig1 = plt.figure()

ax = fig1.add\_axes([0,0,1,1])

ax.bar(np.arange(0,4,0.05), [ele/N/0.05 for ele in pdfa])

ax.set\_title('PDF of Amplitude')

which generates graph like,



Indeed, the graph comes out to a Rayleigh distribution at = have maximum probability density value .

Similarly, in case of we get the graph below, In the following graph it can be seen that it follows nearly uniform distribution with

