

Visualisation

April 18, 2021

1 Data-Visualisation

Visualising Data for Modulation Classification

1.1 Imports

```
[1]: import numpy as np
import matplotlib.pyplot as plt
import os
import scipy.io

from IPython.display import display, Math, Latex, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))
```

<IPython.core.display.HTML object>

1.2 Visualisation

```
[2]: def PlotData(Path,Channel,ModulationType,SNR,L=None):
    S = str(SNR) + "dB-SNR"
    if Channel == "AWGN":
        File = scipy.io.loadmat(Path + Channel + "/" + ModulationType + "/" + S +
        ↪ '.mat')
    elif Channel == "Rayleigh":
        File = scipy.io.loadmat(Path + Channel + "/" + str(L) + "/" +
        ↪ ModulationType + "/" + S + '.mat')

    Data = File['rx']
    TrueData = File['txModulated']

    TrueX,TrueY = TrueData.real, TrueData.imag
    X,Y = Data.real, Data.imag

    N = TrueX.shape[0]
    Ind = np.random.randint(0,N,300)
```

```

plt.figure(figsize=(10,10))
if Channel == "AWGN":
    FileName = "Channel-" + Channel + ", " + "ModulationScheme-" +
    ↪ModulationType + ", " + "SNR-" + str(SNR) + "dB"
elif Channel == "Rayleigh":
    FileName = "Channel-" + Channel + ", " + "ModulationScheme-" +
    ↪ModulationType + ", " + "L-" + str(L) + ", " + "SNR-" + str(SNR) + "dB"
plt.title(FileName)
plt.xlabel("In-Phase Component")
plt.ylabel("Quadrature-Phase Component")
plt.scatter(TrueX[Ind],TrueY[Ind],color='green')
plt.scatter(X[Ind],Y[Ind],color='red')
plt.grid()
plt.savefig("Images/" + FileName + ".jpg")
plt.show()

```

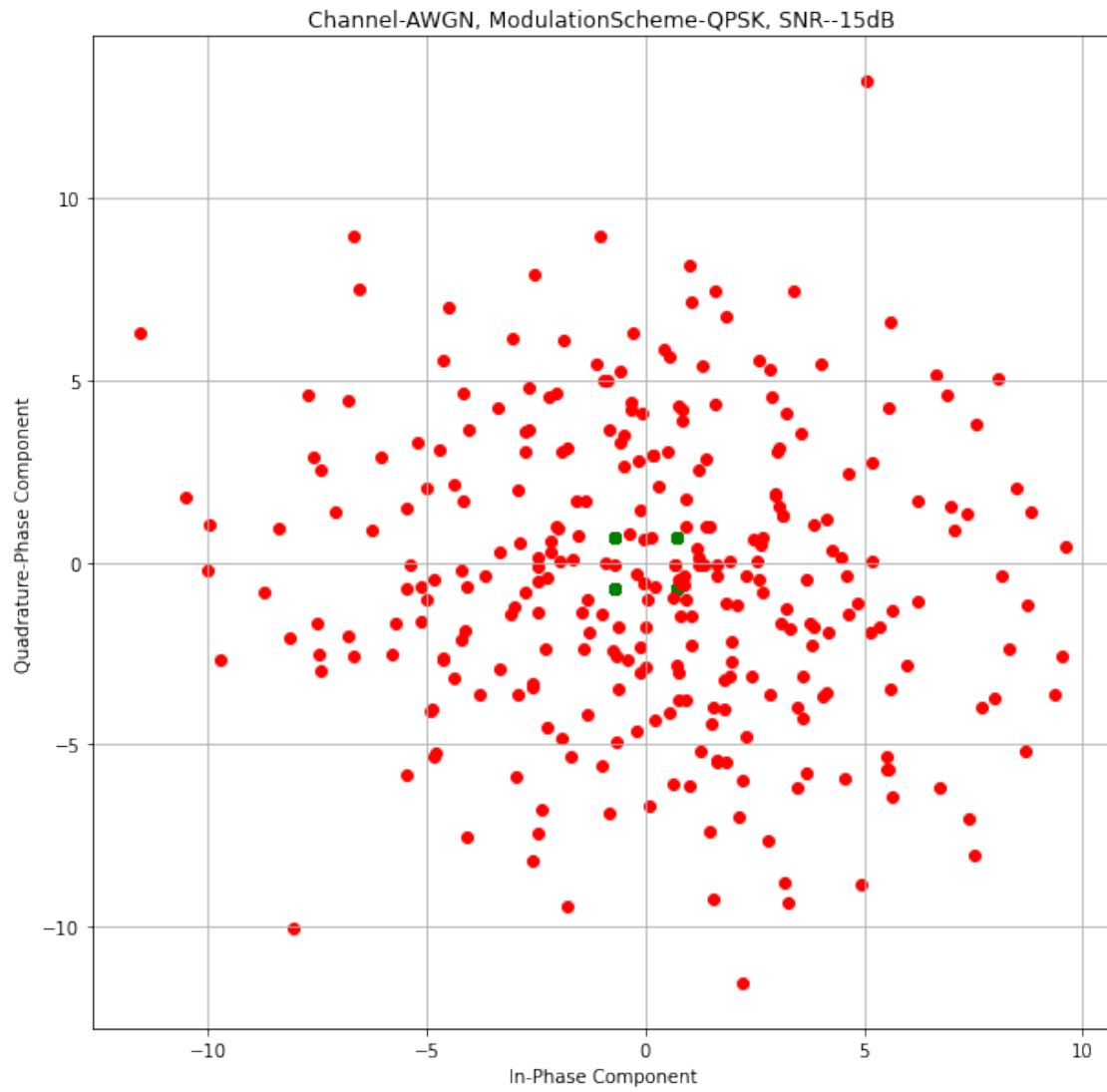
1.2.1 AWGN Channel

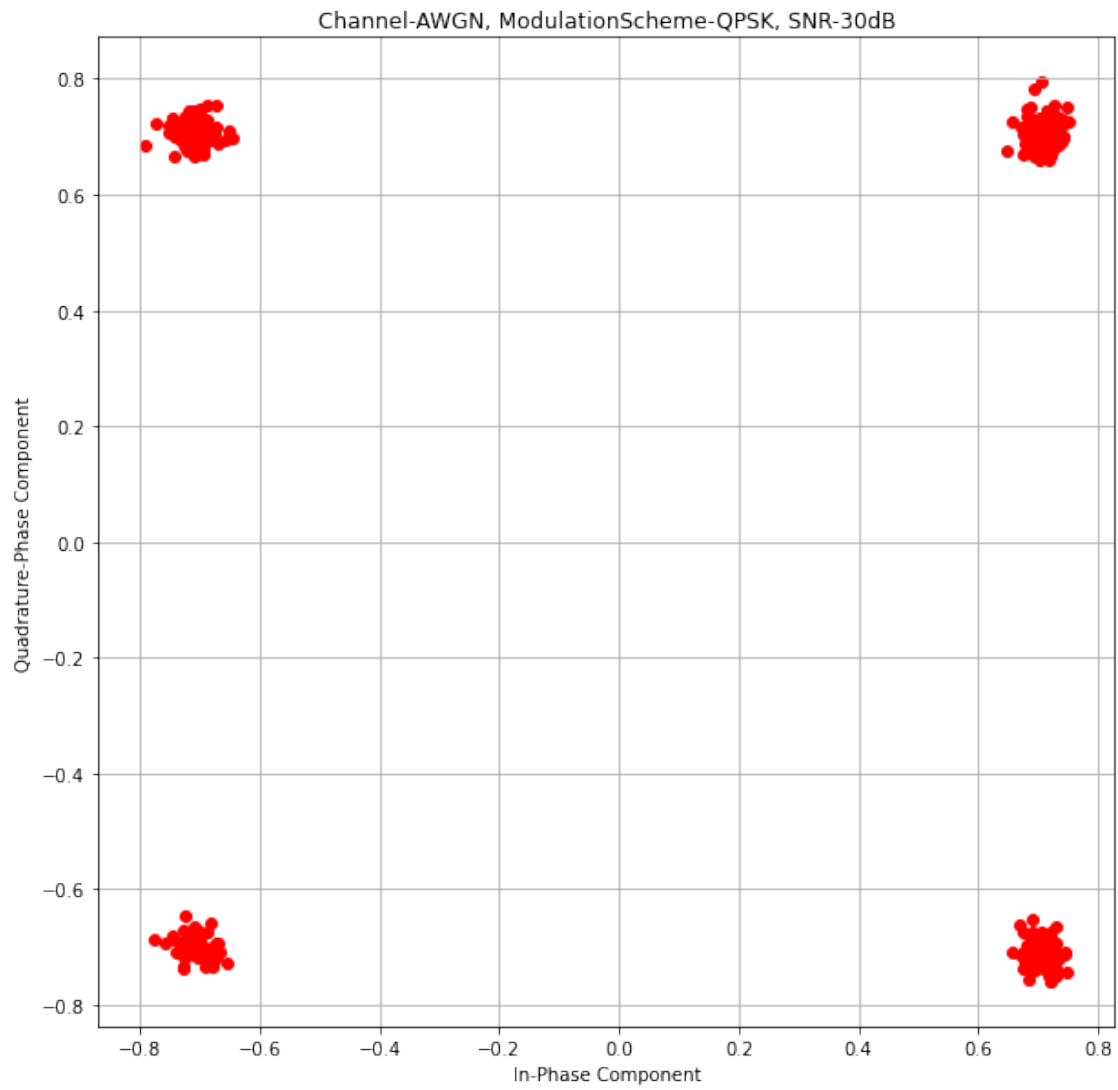
Specifications: - **Green:** True Constellations - **Red:** Constellations after adding Noise

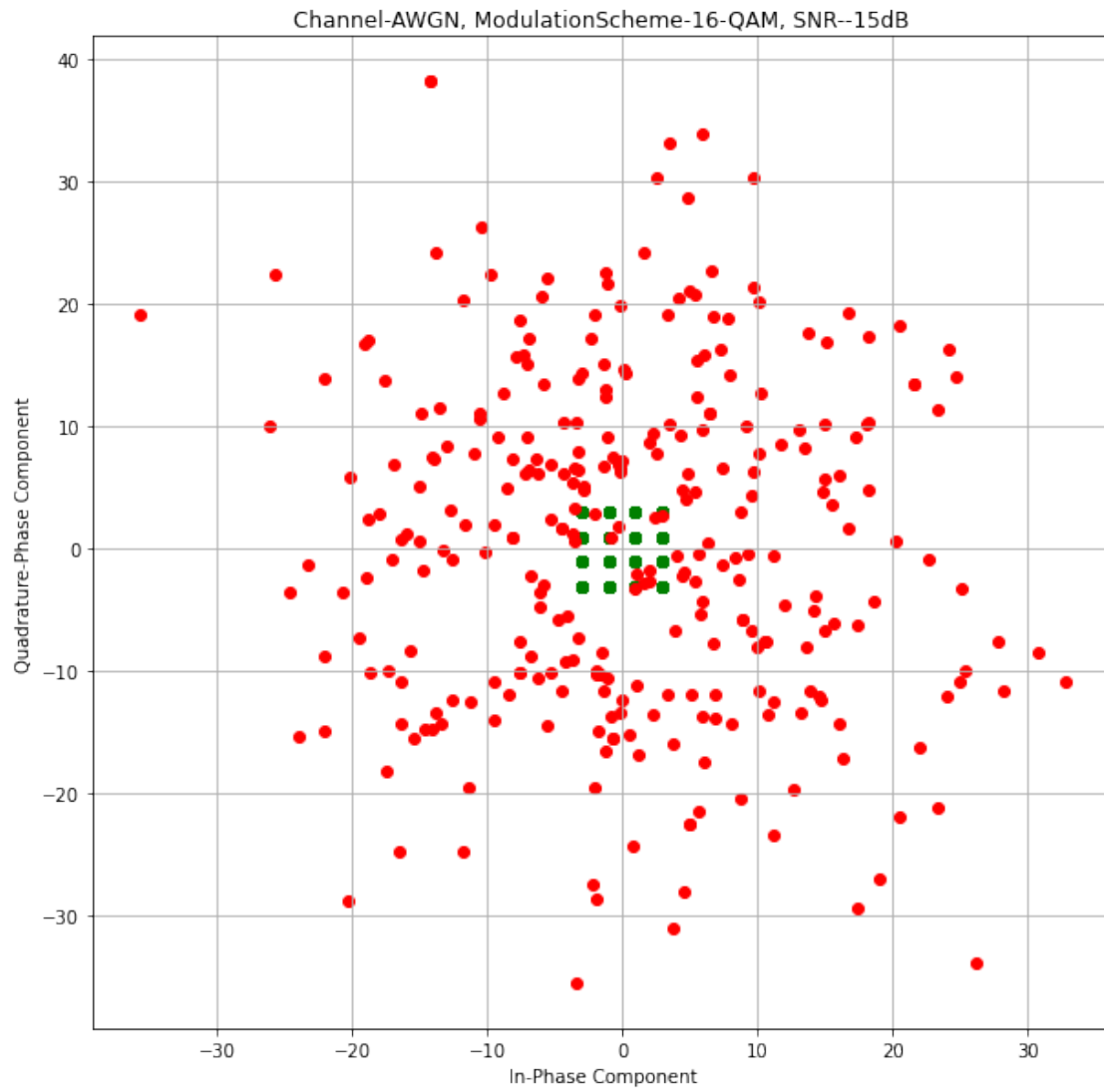
```

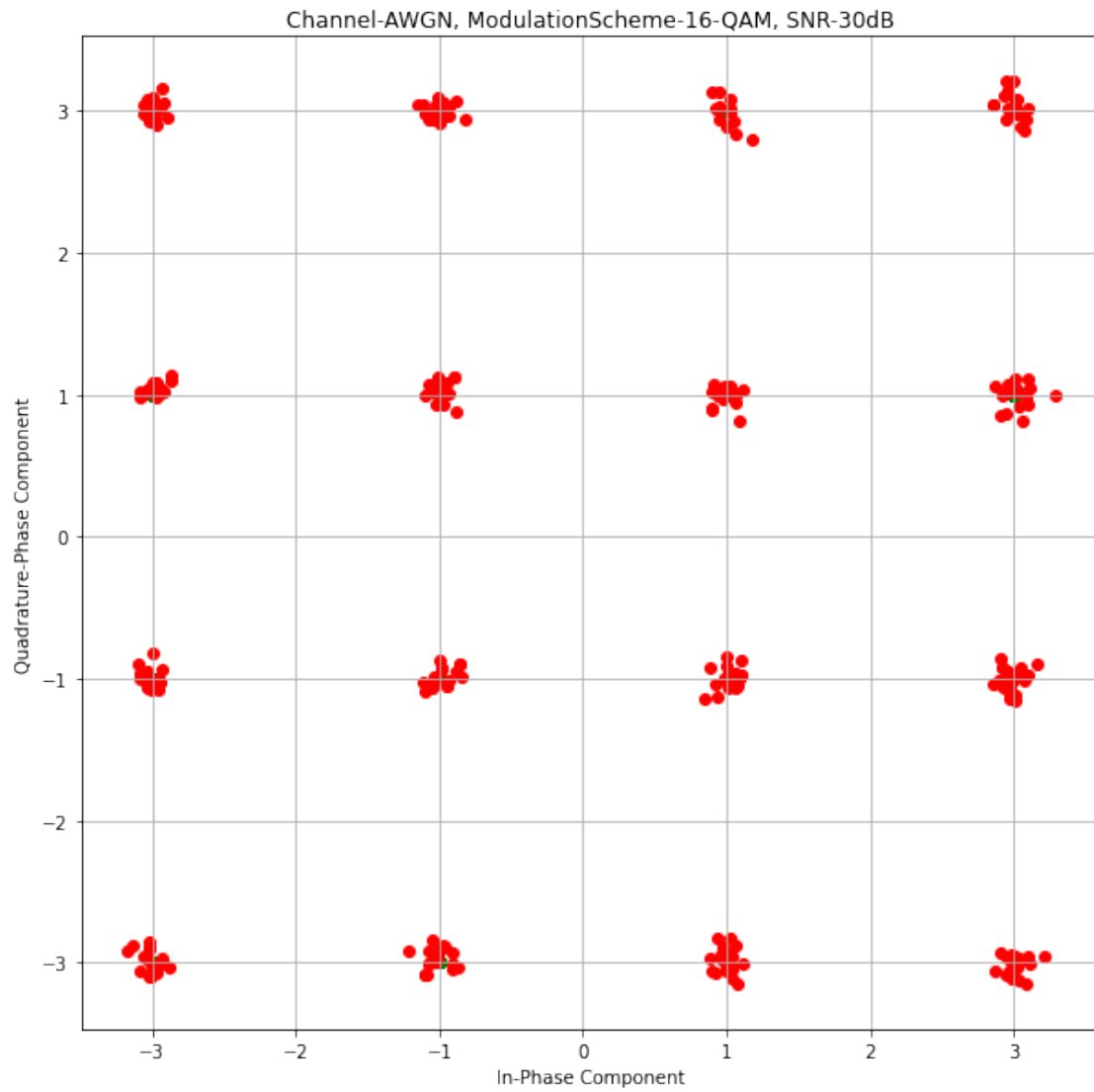
[3]: PlotData("../Data/", "AWGN", "QPSK", -15)
PlotData("../Data/", "AWGN", "QPSK", 30)
PlotData("../Data/", "AWGN", "16-QAM", -15)
PlotData("../Data/", "AWGN", "16-QAM", 30)
PlotData("../Data/", "AWGN", "64-QAM", -15)
PlotData("../Data/", "AWGN", "64-QAM", 30)

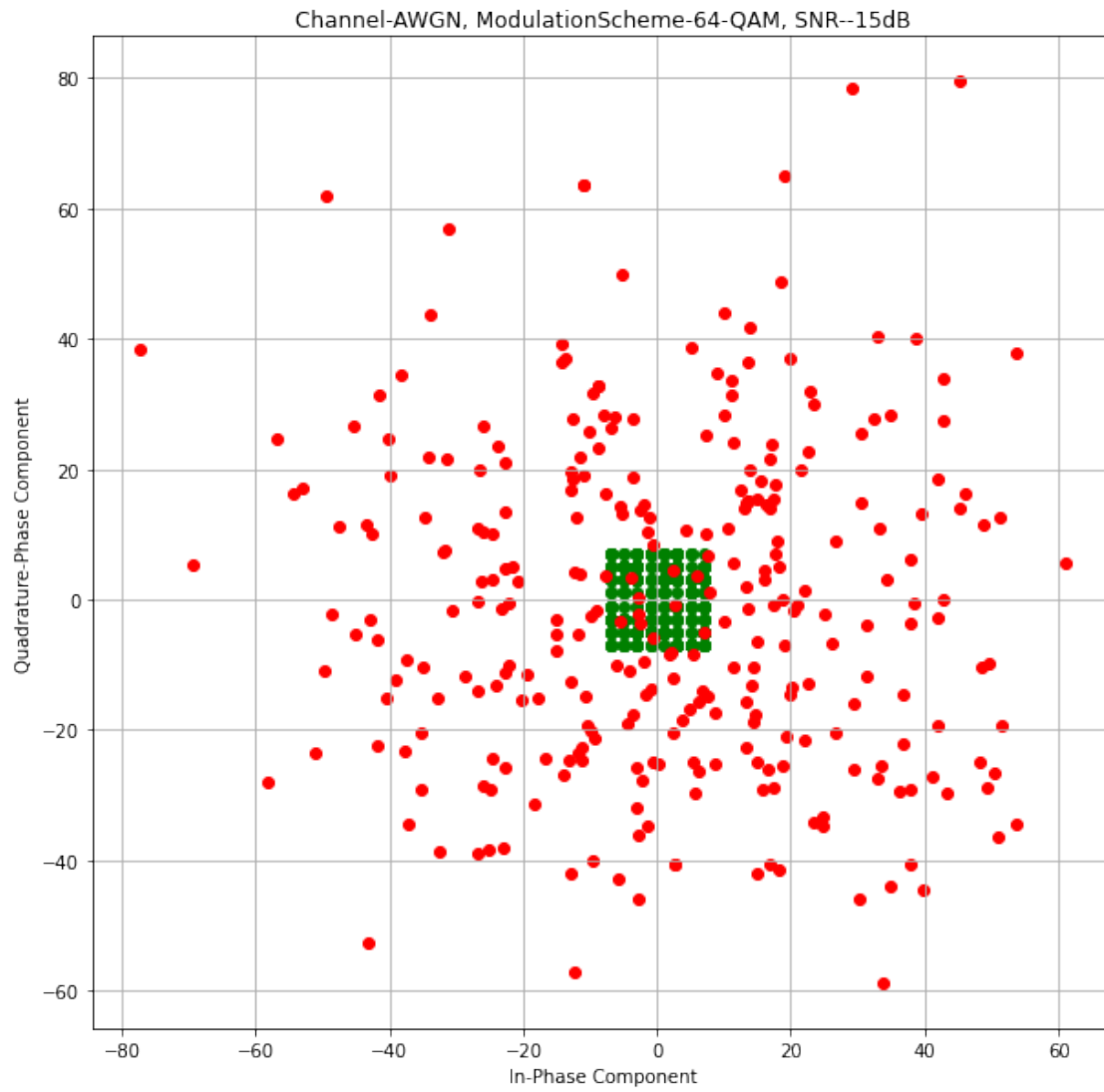
```

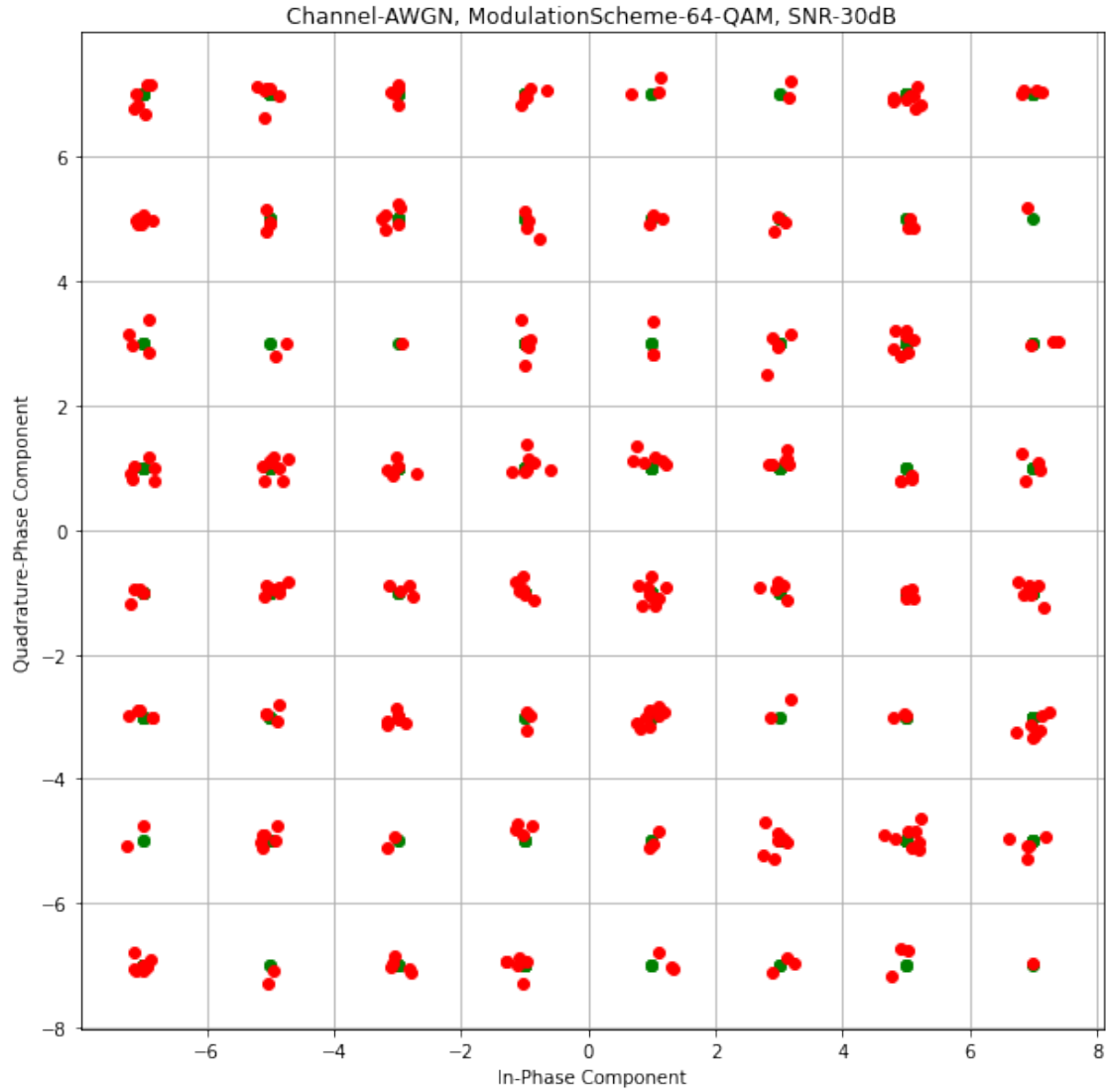








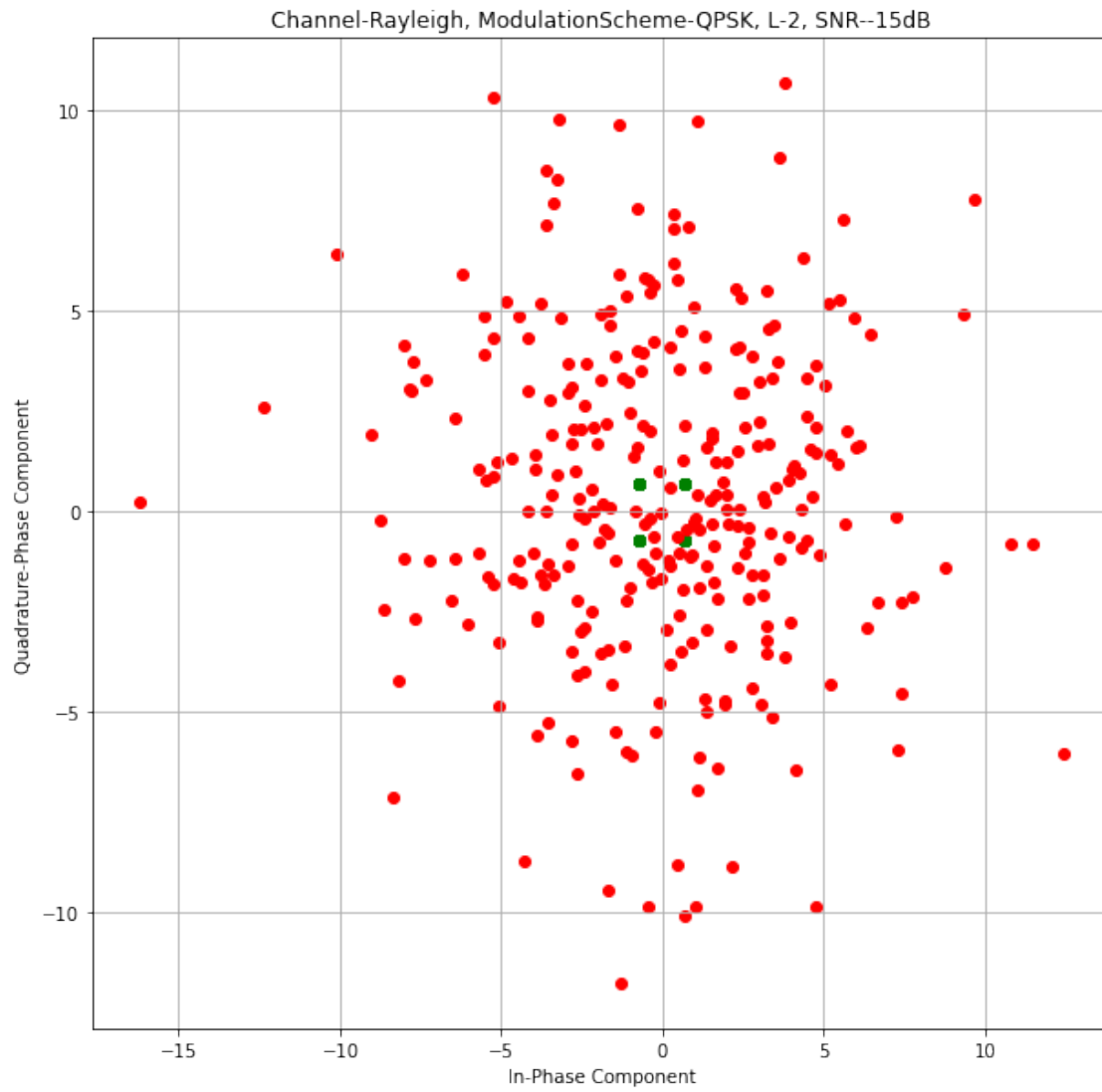


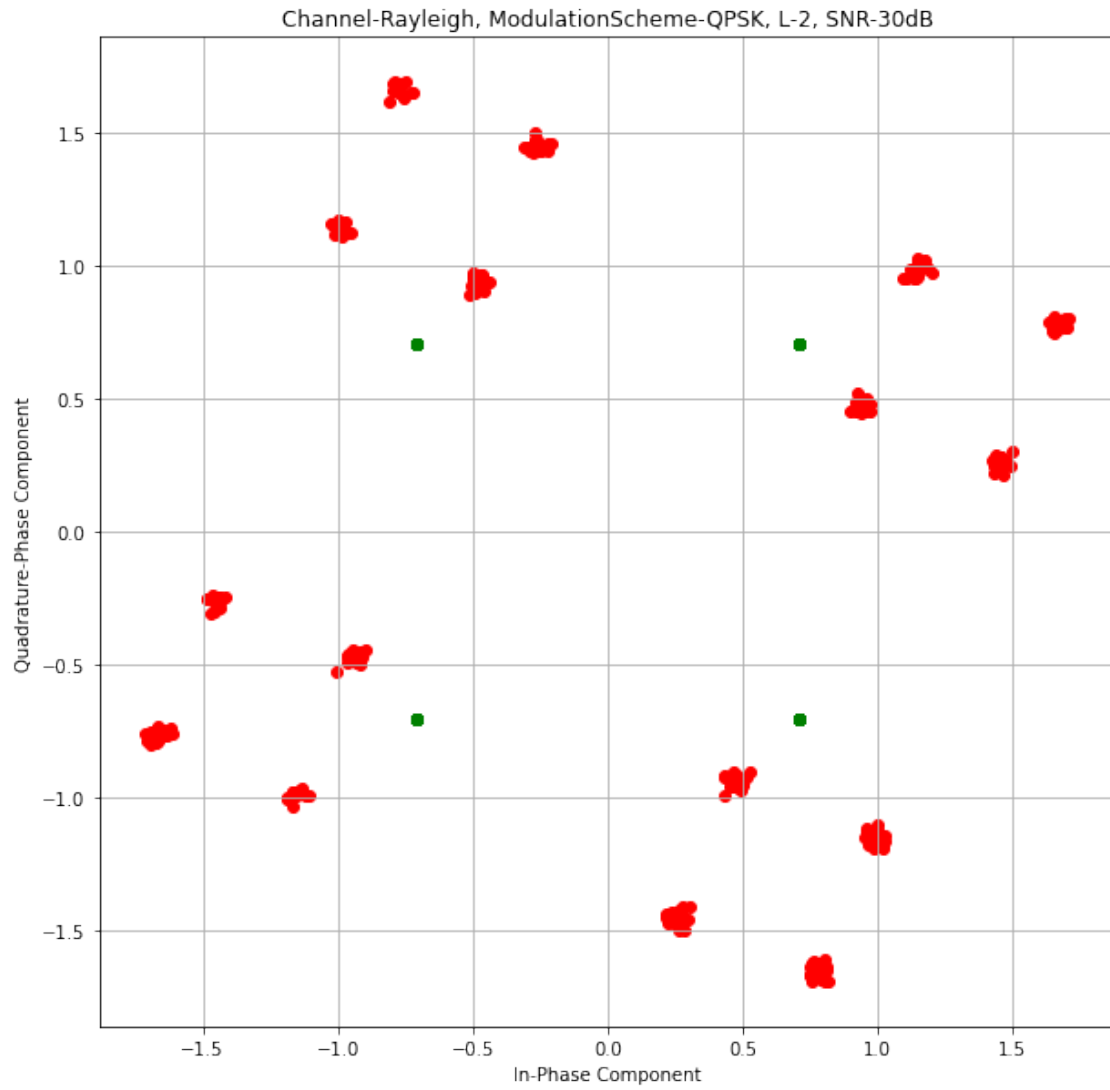


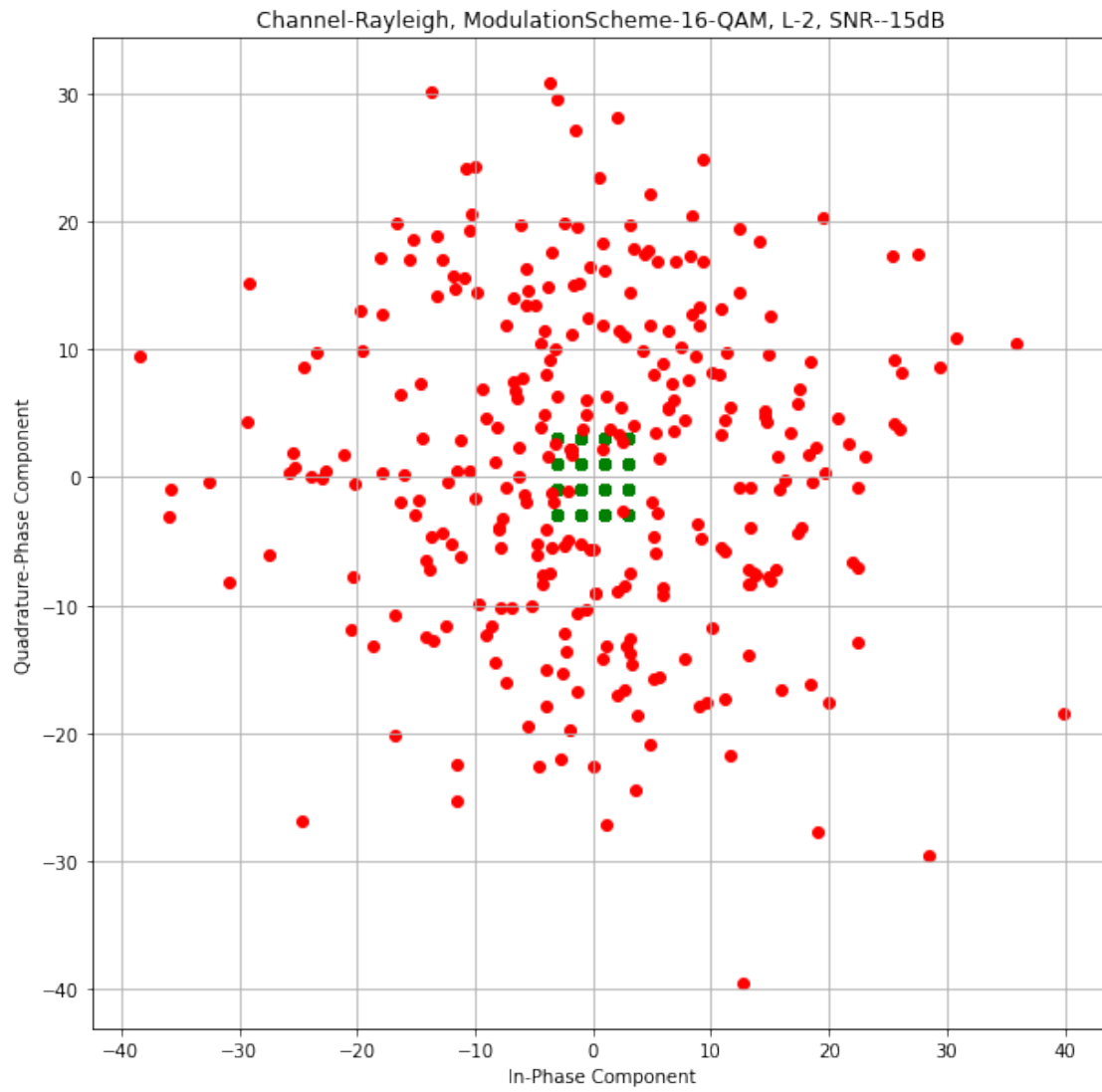
1.2.2 Rayleigh Channel

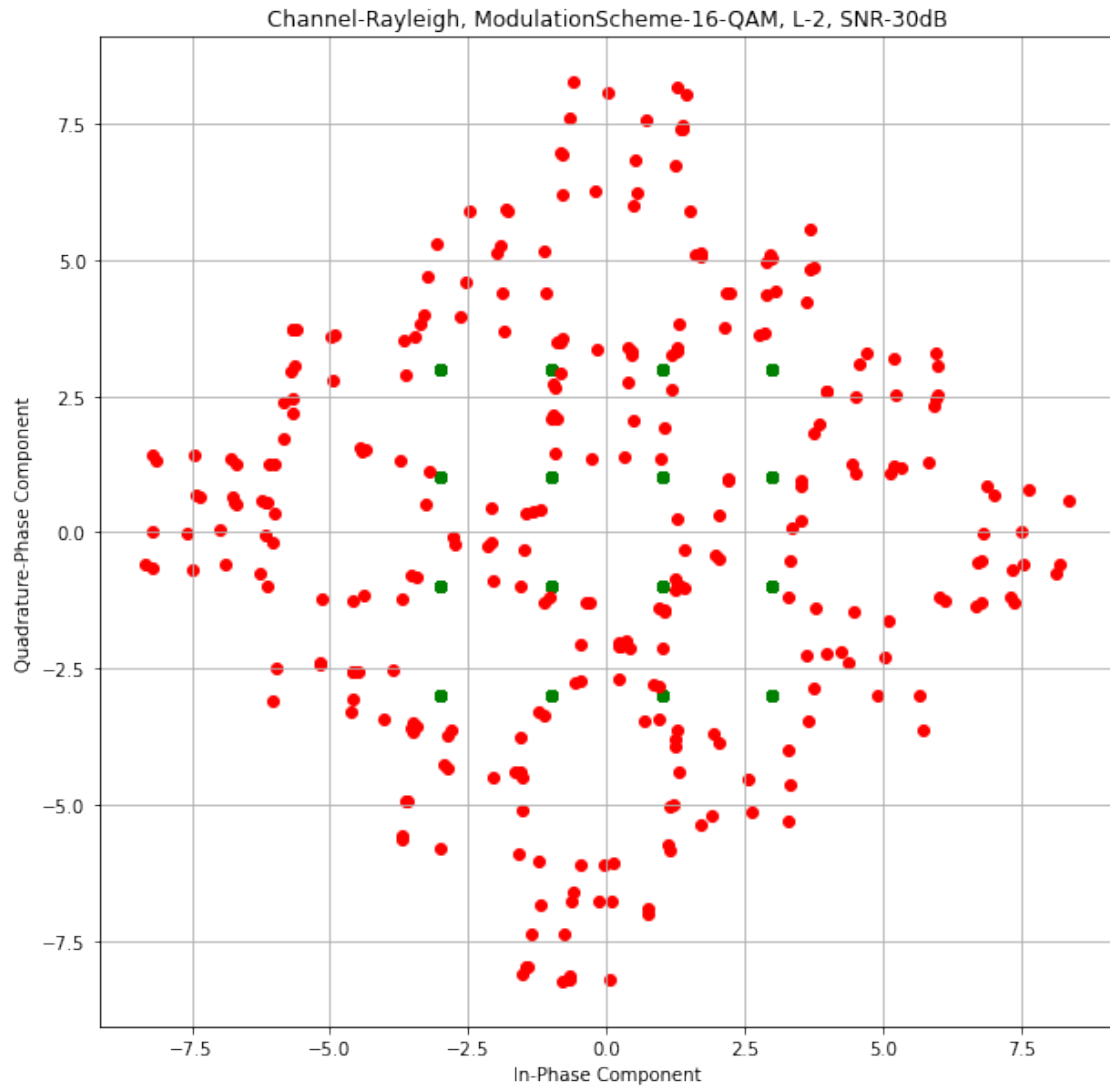
Specifications: - **Green:** True Constellations - **Red:** Constellations after Fading and adding Noise

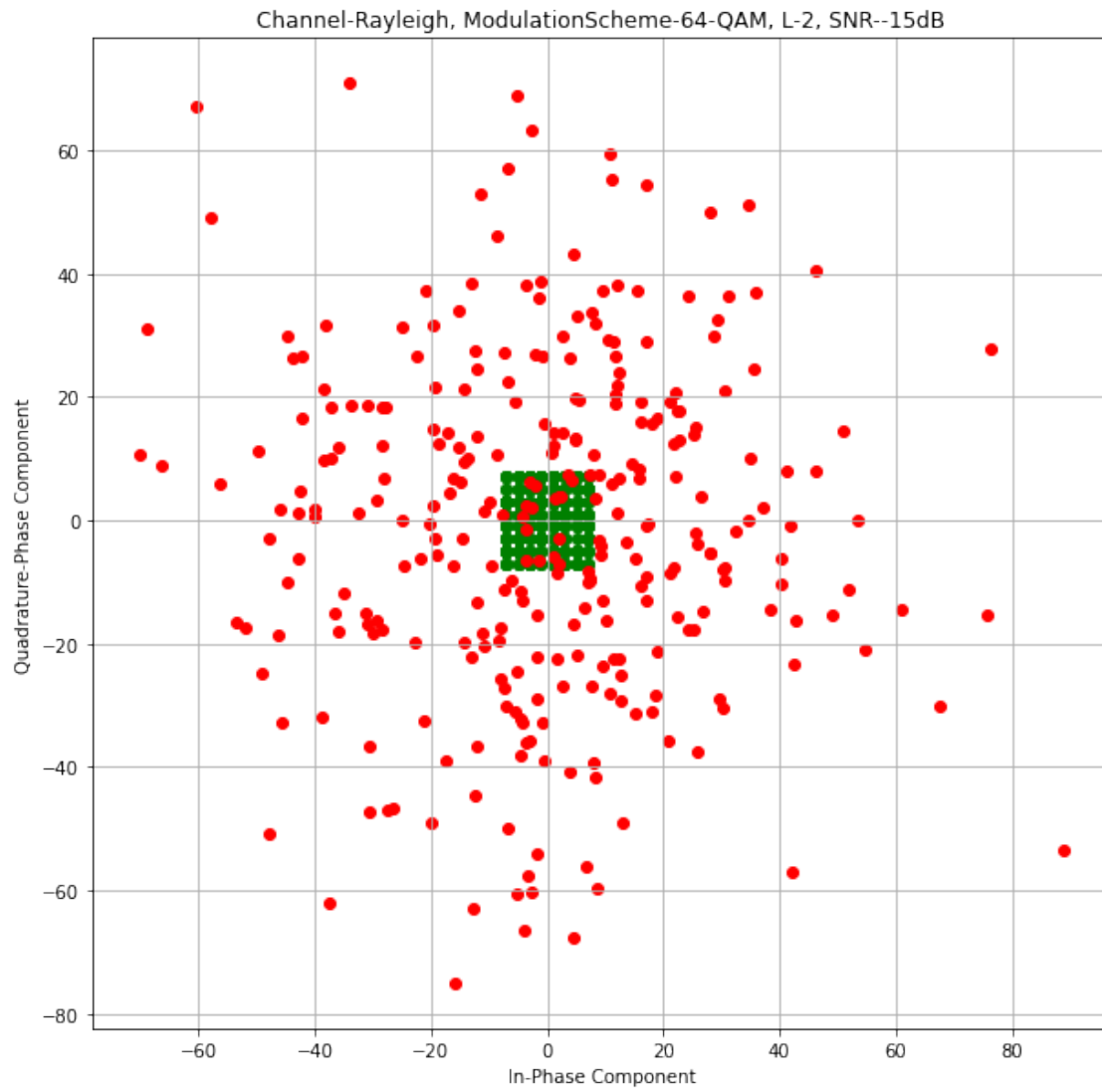
```
[4]: PlotData("../Data/", "Rayleigh", "QPSK", -15, L = 2)
PlotData("../Data/", "Rayleigh", "QPSK", 30, L = 2)
PlotData("../Data/", "Rayleigh", "16-QAM", -15, L = 2)
PlotData("../Data/", "Rayleigh", "16-QAM", 30, L = 2)
PlotData("../Data/", "Rayleigh", "64-QAM", -15, L = 2)
PlotData("../Data/", "Rayleigh", "64-QAM", 30, L = 2)
```

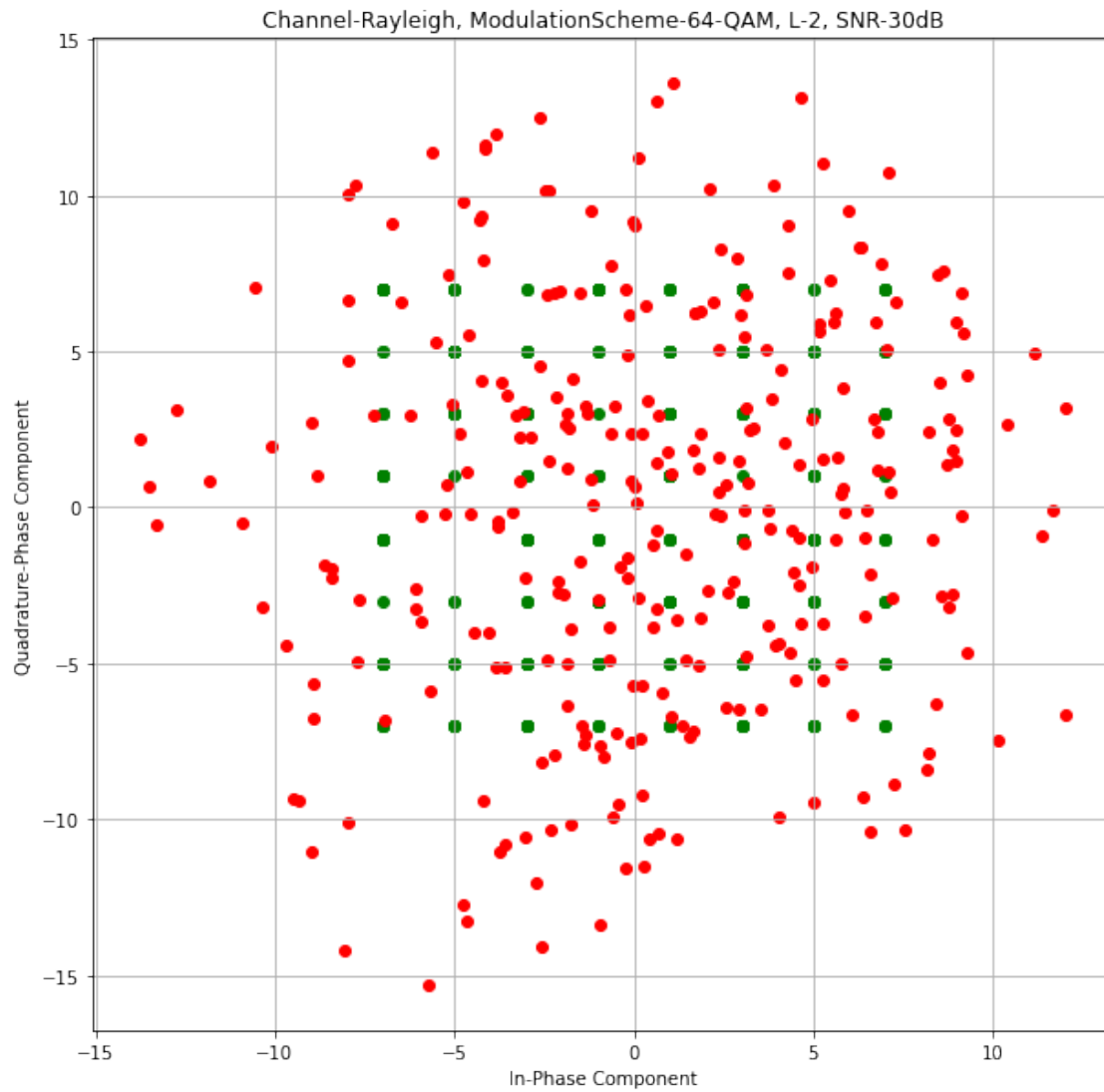













```
[5]: PlotData("../Data/", "Rayleigh", "QPSK", -15, L = 3)
PlotData("../Data/", "Rayleigh", "QPSK", 30, L = 3)
PlotData("../Data/", "Rayleigh", "16-QAM", -15, L = 3)
PlotData("../Data/", "Rayleigh", "16-QAM", 30, L = 3)
PlotData("../Data/", "Rayleigh", "64-QAM", -15, L = 3)
PlotData("../Data/", "Rayleigh", "64-QAM", 30, L = 3)
```

