

Data Generation : Modulation Classification

EE18BTECH11014

Krishna Srikar Durbha

Signal and Channel Parameters

```
% No.of Samples
N = 10000;

% SNR(in dBW)
SNR = [5,10,15,20,25,30];

% Modulation Schemes
modulationTypes = categorical(["QPSK", "16-QAM", "64-QAM"]);

% Channels
channelTypes = categorical(["AWGN", "Rayleigh"]);
```

Data Generation

Generating Signal

```
numModulationTypes = length(modulationTypes);
numChannelTypes = length(channelTypes);
for i = 1:numChannelTypes
    for j = 1:numModulationTypes
        if channelTypes(i) == "Rayleigh"
            DataGeneration(N*100,modulationTypes(j),channelTypes(i),SNR)
        elseif channelTypes(i) == "AWGN"
            DataGeneration(N,modulationTypes(j),channelTypes(i),SNR)
        end
    end
end
```

```
Saved AWGN QPSK Data
Saved AWGN 16-QAM Data
Saved AWGN 64-QAM Data
Saved Rayleigh QPSK Data
Saved Rayleigh 16-QAM Data
Saved Rayleigh 64-QAM Data
```

Functions

Bit Error Rate Calculation for a Signal

```
function DataGeneration(N,Modulation,Channel,SNR)

    if Modulation == "QPSK"
        tx = randi([0 3], N, 1);
    elseif Modulation == "16-QAM"
```

```

        tx = randi([0 15], N, 1);
elseif Modulation == "64-QAM"
    tx = randi([0 63], N, 1);
end

% File Path
dataDirectory = fullfile("../Data/" + string(Channel) + "/" + string(Modulation) +
mkdir(dataDirectory);

% Modulation: Modulating Data and Scatter Plotting it
if Modulation == "QPSK"
    qpskmod = comm.QPSKModulator;
    txModulated = qpskmod(tx);
    %scatterplot(txModulated);
    %grid on;
elseif Modulation == "16-QAM"
    txModulated = qammod(tx,16);
    %scatterplot(txModulated);
    %grid on;
elseif Modulation == "64-QAM"
    txModulated = qammod(tx,64);
    %scatterplot(txModulated);
    %grid on;
end

S = size(SNR,2);

% Transmission: Transmission of Data through Channel and
% Decoding: Decoding the Received Data

for i = 1:S
    snr = SNR(i);
    % Fading and Noise
    if Channel == "Rayleigh"
        % disp(string(Channel) + " " + string(Modulation) + " " + string(snr))
        rayleighchan = comm.RayleighChannel();
        tx = rayleighchan(tx);
        rx = awgn(tx,snr);
    elseif Channel == "AWGN"
        % disp(string(Channel) + " " + string(Modulation) + " " + string(snr))
        rx = awgn(txModulated,snr);
    end

    % Save data file
    fileName = fullfile(dataDirectory,sprintf("%sdB-SNR",string(SNR(i))));
    save(fileName,"tx","txModulated","rx","snr");
end
disp("Saved " + string(Channel) + " " + string(Modulation) + " Data")
end

```