Task 1

Perform the process of encode / modulate / transmit HEX characters over a transmit channel with Gaussian noise addition / demodulation / decoding for a random data file size of 3000 HEX characters as below:

The character space is as follows:

0=0 → 0001

(…)

1 2 3 4 5 6 7 8 9 A B C D E

(…)

F=15 → 1111

Compare the error rate of the above-mentioned characters (expressed in%) of transmitting 3000 HEX characters obtained at the output of this process for 16-QAM modulation, assuming for SNR in the range of 2, up to 30 dB. Graph the results of the message transmission error rate for all tasks using the script lab1.m

Readings

1. https://en.wikipedia.org/wiki/Quadrature_amplitude_modulation
2. https://www.mathworks.com/help/comm/gs/examine-16-qam-using-matlab.html

Solution hint

You can use the function:

- `rng` — Controls the random number generation.
- `randi` — Generates a random binary data stream.
- `bi2de` — Converts the binary signal to an integer-valued signal.
- `qammod` — Modulates using 16-QAM.
- `comm.AWGNChannel` — Impairs the transmitted data using AWGN.
- awgn - Add white Gaussian noise to signal
- `scatterplot` — Creates constellation diagrams.
- `qamdemod` — Demodulates using 16-QAM.
- `de2bi` — Converts the integer-valued signal to a binary signal.
- `biterr` — Computes the system BER.

Here, you will find a solution hint:

```
%% generate n-random characters
n = 500;
x = randi(n,1,16);  % input signal
%% QAM modulation
y = qammod(x,16); % signal generation for QAM16 modulation
snrVal = 30;
y_noise = awgn(y, snrVal,'measured'); % add white Gaussian Noise SNR 30 dB, SNR, ang. signal-to-noise ratio) - a measure that compares the level of the useful signal (information) to the level of background noise (unwanted signal, expressed in the decibel [dB] unit)

%% demodulation
z = qamdemod(y_noise,16); % signal demodulation-> output signal

%% data visualization as a scatter plot -> https://en.wikipedia.org/wiki/Quadrature_amplitude_modulation
sy = scatterplot(y,1,0,'b*');
hold on;
scatterplot(y_noise,1,0,'g.',sy);
legend('Signal Constellation','Received Signal');
hold off;
%% error calculation
Error1 = z-x; % non zero is error
```

In order to get information about a given function you should use

help *function_name*

In order to obtain a vector with the calculated Error1 for different snrVal values you should use a loop:

```
snrVal = [0:5:30];
% snrVal = [-30:5:0]; % a vector with negative values

for i = 1:length(snrVal)
  y_noise (i) = awgn(y, snrVal(i),'measured');
      z(i) = qamdemod(y_noise(i),16);
  error1(i) = z(i)- x;
end
```

In order to visualize the data in a plot, please use plot function

```
figure, %it opens the graphical window
plot(snrVal,error1,'r-') %where plot(vector of X,vector of Y, line style)
title('error of QAM16'); %create title of plot
xlabe('SNR value'); %create name of X vector
ylabel('error in [%]'); % create name of Y vector
```

In order to present two plots in one figure window, please use:

```
plot(x1,y1,'r-',x2,y2,'g--');
```