

Unit 4. In-Class Exercises

These exercises are to be performed with the in-class exercises.

Problem 1: Theoretical Error Rate Probability

Modify the code in the demo to compute the BER vs. E_s/N_0 for 16-QAM for AWGN and fading channel. (Recall the demo measured the BER vs. E_b/N_0). Remember that you will want to plot the BER with semilogy.

```
% Parameters
bitsPerSym = 4; % 4=16-QAM
nsym = 10000;

% TODO
%   ber = ...
```

We will now compare the measured BER with the theoretical value. For 16-QAM, the theoretical BER for an AWGN channel is

$$\text{BER} = 3/4 * \text{qfunc}(\sqrt{E_s/N_0})$$

Use the Lemma in the lecture notes, to also compute the theoretical BER for a fading channel. Plot the theoretical and measured values.

```
% TODO:
%   berTheoryAwgn = ...
%   berTheoryFading = ...
```

Problem 2: Outage Capacity in an Indoor Environment

In this problem, we will estimate the outage capacity in an indoor setting. Our goal is to look at the effects of both large-scale and small-scale fading.

First generate n_x locations in a box of size 30 x 40 m representing locations in some large indoor environment. Assume an access point is located at the origin and compute a vector, `dist`, representing the distance in meters from the AP to each location.

```
% Parameters
nx = 10000;
xmax = [30,40];
d = 2;

% TODO
%   x = ...
%   dist = ...
```

Next, use the `pathLoss3GPPInH` function below to compute random path losses for each location. Based on the path losses and the parameters below compute the vector, `snrAvg` representing the average SNR for each location. Plot the CDF of the SNRs.

```
% Parameters
fc = 2.3e9; % Carrier freq in Hz
ptx = 15; % TX power in dBm
nf = 10; % Noise figure in dB
bw = 16e6; % Bandwidth in Hz
Ekt = -174; % Thermal noise in dBm/Hz before NF

% TODO
% pl = pathLoss3GPPInH(...);
% snrAvg = ...
```

For each average SNR, `snrAvg(i)`, generate a random instantaneous SNR, `snr(i)`, representing the average SNR with Rayleigh fading. Plot the CDF of `snr` and `snrAvg`.

```
% TODO
% snr = ...
```

For each SNR, assume the rate is given by the function `rate` below. Plot the CDF of the rate for the average SNR and instantaneous SNR. Note:

- `rate(snrAvg, bw)`: Variation in rates with only large-scale variations
- `rate(snr, bw)`: Variation in rates with large-scale variations and small scale fading

```
rate = @(snr, bw) bw*min(4.5, 0.6*log2(1+db2pow(snr)));

% TODO
% rateAvg = ...
% rateFading = ...
```

Problem 3: NR LDPC Coding

In this problem, we will simulate the LDPC encoding and decoding used in the 5G NR standard. The MATLAB 5G Toolbox has an amazingly good implementation of this code, so we can just call it. In the NR LDPC code, the number of input bits is given by:

```
nbitsIn = nrows*nlift;
```

where `nrows` is the number of rows in the LDPC base graph, and `nlift` is the so-called lifting factor which expands the graph to different block sizes. We will use the following parameters:

```

bgn = 1;      % LDPC base graph number (1 or 2)
nrows = 22;  % number of rows in the base graph
nlift = 128; % lifting factor
nbitsIn = nrows*nlift; % number of input bits
maxNumIters = 8;      % max number of LDPC decode iterations

```

Using MATLABs nrLDPCencode and nrLDPC decode functions to simulate the NR LDPC coding. Using code similar to the demo, estimate the block error rate as a function of the SNR for Eb/N0 values in EbN0Test.

```

% SNR values to test
EbN0Test = (0:0.25:3)';
ntest = length(EbN0Test);

% TODO: Estimate BLER at each Eb/N0
% bler = ...

```

```

% TODO: Plot BLER vs Eb/N0

```

```

function pl = pathLoss3GPPInH(dist,fc)
    % pathLoss3GPPInH: Generates random path loss
    %
    % Samples the path loss using the 3GPP-InH model

    % Compute the median path losses for LOS and NLOS
    pllos = 32.4 + 17.3*log10(dist) + 20*log10(fc/1e9);
    plnlos = 17.3 + 38.3*log10(dist) + 24.9*log10(fc/1e9);

    % Add shadowing
    w = randn(size(dist));
    pllos = pllos + 3*w;
    plnlos = plnlos + 8.03*w;

    % Compute probability of being LOS or NLOS
    plos = min( exp(-(dist-1.2)/4.7), 1);
    plos = max( 0.32*exp(-(dist-6.5)/32.6), plos);

    % Select randomly between LOS and NLOS path loss
    u = (rand(size(dist)) < plos);
    pl = u.*pllos + (1-u).*plnlos;

end

```