Spring 2022 – Cooperative Communications and Networking

Assignment 2

Instructor: Kuang-Hao (Stanley) Liu

Numerical Questions

Q1) [Outage probability, 40%] Derive the outage probability of selection combining (SC) for N receive antennas over Rayleigh fading channels. Let highest SNR is given by $\gamma_{\max} = \max_{n=1,\cdots,N}\left(|\alpha_n|^2\frac{E}{N_0}\right)$

Hint: Denote $X_{\max}$ as the maximum of N i.i.d. random variables $X_1, \cdots, X_{\max}$. The cumulative distribution function (CDF) of $X_{\max}$ is given by

$$F_X(x) = \mathbb{P}[\max(X_1, \cdots, X_N) < x] = \prod_{n=1}^{N} \mathbb{P}[X_i < x].$$

Q2) [High-SNR approximation, 20%] The outage probability derived in Q2 is a function of the average SNR of each branch. If the average SNR of the branches are the same and approaching to infinity, show that the outage probability is proportional to the average SNR to the power of $-N$.

Hint: Use the first-order Taylor series $e^{-x} \approx 1 - x$.

Simulation

Q3) [SC, 40%]

Perform Monte Carlo simulation for the outage probability of selection combining (SC).

- Consider the average SNR from 0 dB to 15 dB, the number of receive antennas $N = 1, 2, 3$, the decoding threshold $T = 1$.
- (25%) Plot both the theoretical outage probability derived from Q2 and the simulated ones versus SNR in the same figure. Full credits are given when the following requirements are all satisfied.
  - Clearly indicate the value of N for each curve and whether it is a theoretical curve or a simulated one.
  - Set y-axis as logarithm scale.
  - Colors are not differentiable when printed in black and white. Use different line styles and symbols to represent different curves.
- (10%) Explain your simulation program by plotting a flow chart.
- (5%) Explain how you determine the diversity gain of each curve.

A sample code that simulates a point-to-point transmission (without relays) over Rayleigh fading channel is provided below.

——————————————

```
///// Outage probability simulation over Rayleigh fading channel /////

// Class definition
class glb:
    snrdBmin = 0
    snrdBmax = 15
    snrdBinterval = 1
    T = 1 // threshold
    MaxSamples = pow(10,6) // max number of runs
    n_stop = pout_ana*(1-pout_ana)*pow(1.96/ pow(10,-4), 2) // num. of required runs

while (not Stop):
           n_0 = 1/math.sqrt(2)*(np.random.normal(0,1)+np.random.normal(0,1)*1j) // white
               Gaussian noise
           h_0 = 1/math.sqrt(2)*(np.random.normal(0,1)+np.random.normal(0,1)*1j) // Rayleigh
               channel coefficient

           // compute SNR
           snr_0 = abs(h_0)**2/pow( abs(pow(10,-snrdB/20.0)*n_0), 2)

           // count the errors
           if snr_0 < glb.T:
               n_err += 1

           n_s += 1 // number of current runs
           // stop criterion
           Stop = (n_s > n_stop) or (n_s > glb.MaxSamples)
```