Monte Carlo Simulation - MATLAB analysis

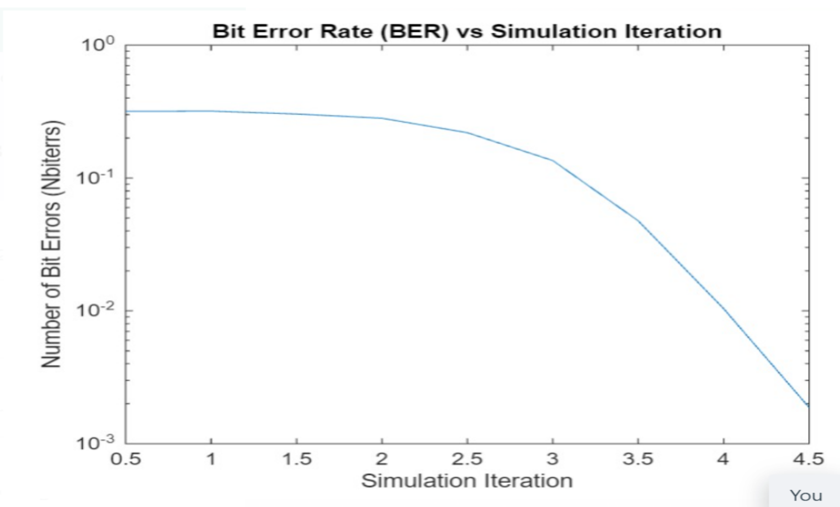
We performed the Monte-Carlo simulation for each Eb/N0 ∈ {0.5, 1, 1.5, 2.5,3.5 ….10} dB.

We plotted a graph of Bit Error Rate (BER) (in log scale) vs SNR values.

For each SNR values from 0.5 to 10, we performed ‘Nsim’ simulations where ‘Nsim’ values varied from 100 to 10000 for different observations. For e.g. we plot the graph for Nsim=100, Nsim=500, Nsim=1000, Nsim=10000.
We tallied the number of bits of decoded vector which doesn’t match with the input sequence, since this is the total number of bit errors in this simulation. Hence, to double check it, we also printed the number of bit errors as well as the decoded vector and input sequence for each SNR values from 0.5 to 10 at the output window in MATLAB.

In each plot (Nsim 100 to 10000) , we observed the shape of the graph similar as ‘Waterfall’ shape as expected, because as the SNR value increase, the received signal strength is much stronger than the noise strength and the Bit errors decrease rapidly after some point of SNR value.

MATLAB Graph (Observed) : Y-axis ; Bit error rate (in log scale) X-axis ; SNRvalue



Command Window

```
Enter the value of N:
512
Enter the value of K:
340
```

SNR Value	BER
0.5	0.3175
1	0.3188
1.5	0.3034
2	0.2818
2.5	0.2194
3	0.1353
3.5	0.0476
4	0.0104
4.5	0.0019

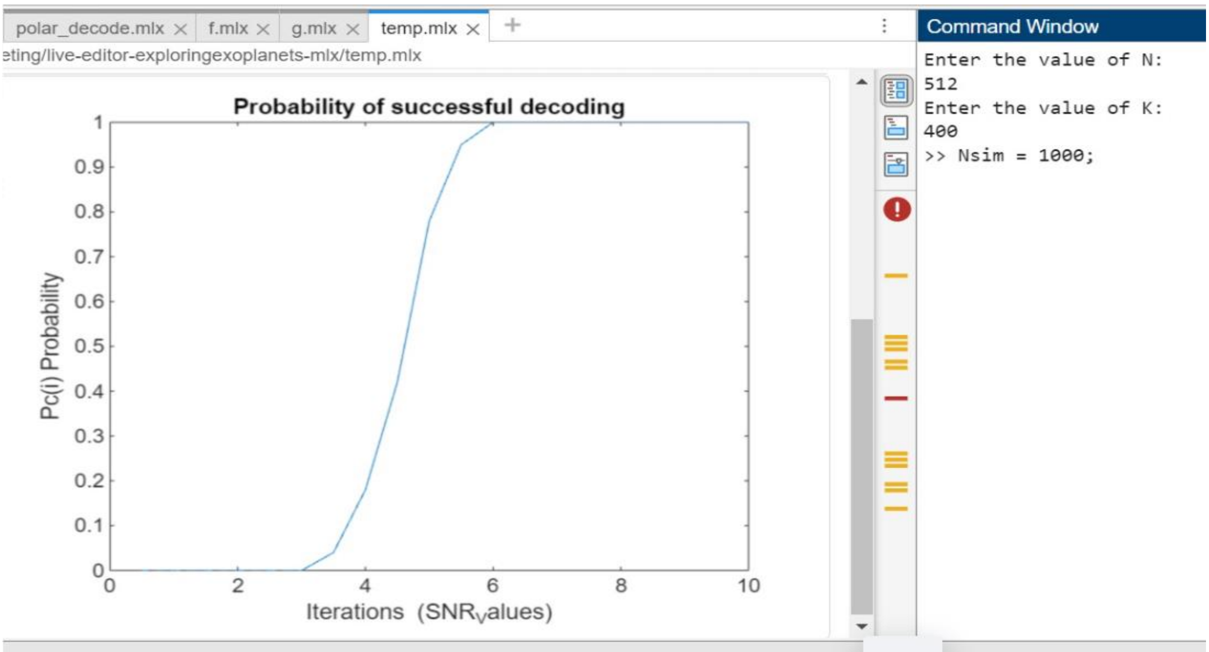
Probability of Decoding success in MATLAB

-> In order to determine the probability of decoding success, for each SNR value, i.e 0.5 to 10 with a step of 0.5 , we set a flag F=0 , before iterating through 1000 simulations. At each iteration, from 1 to 'Nsim'=1000 , if the decoded sequence matched correctly with the input sequence , we set the Flag to F=1, else to F=0.

 Algorithm Convergence: We calculated the no. of times the flag was set to 1 (Success count) during Nsim simulations. We then divided it by Nsim to get the Probability :

$$P_c(i) = \frac{1}{N_{sim}} \sum_{n=1}^{N_{sim}} F_{n,i}.$$

We obtained the plot in MATLAB as follows : N=512 , K=400 and Nsim=1000



The plotted results show that the probability of decoding success increases as SNR increases. This aligns with expectations, as higher SNR generally means less noise and better decoding performance.

We can also relate it with the Bit Error Rate (BER) and Frame error rate (FER) as the BER and FER were high for low SNR values. This suggests that errors in the decoded sequence are more prevalent when the SNR is low, leading to lower decoding success probabilities.

Successive Cancellation List Decoding

For each decoded sequence produced after the successive cancellation decoding , we pass that sequence for CRC check. CRC check is a widely used algorithm for error detection purposes.
So, if the decoded sequence doesn't pass the CRC check, we won't accept that received sequence.

In the successive cancellation decoding, we recursively made decisions for decoding based on the belief, i.e if L(u)>0 u_estimator=0 and if L(u)<0 then u_estimator =1. Now, these decisions may be erroneous.

So, in order to tackle this, we us successive cancellation list decoding which stores the rejected decisions (as decision metric), which may be the correct ones.
We store the mentioned values in the vector as decision metric (DM) for each bit decoding.
if $L(u_i) \geq 0$: $\hat{u}_i = 0$ has $DM_i = 0$, $\hat{u}_i = 1$ has $DM_i = |L(u_i)|$
if $L(u_i) < 0$: $\hat{u}_i = 1$ has $DM_i = 0$, $\hat{u}_i = 0$ has $DM_i = |L(u_i)|$
If the decoded sequence fails the CRC check, then we select the path with least path metric, where path metric is the sum of all DMs in that path.

We implemented it in MATLAB and the BER vs SNR plot is as follows :