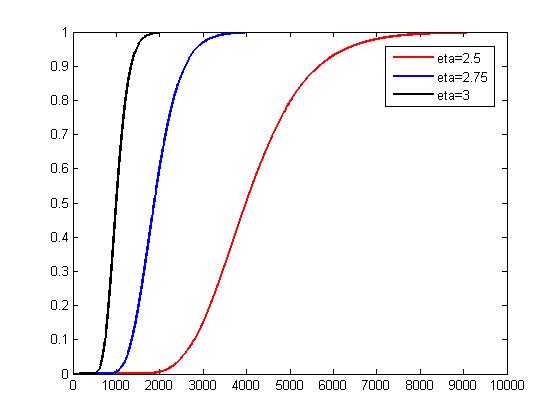
EE597 HW 2

**1.**

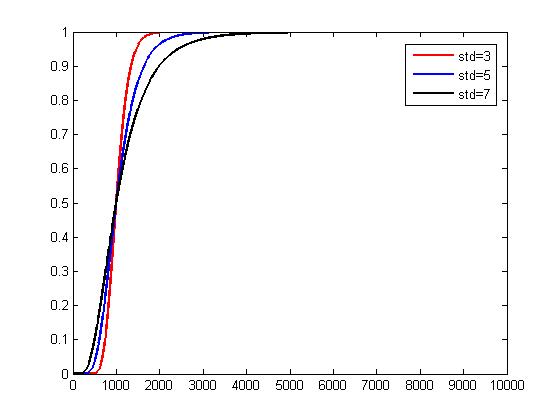
Where, ~.

Comment: as we increase the path loss exponentηvalue, the curve will be shifted to the left. Smallerηvalue allow us to transmit the signal much further without outage. The smaller standard deviation σ value will make the transition part of the curve increase much more dramatically. We can say that when the distance is under 1000m, smaller std is better, while the distance is above 1000m, larger std is better.

std = 3, eta = 2.5, eta = 2.75, eta = 3



eta = 3, std = 2.5, std = 2.75, std = 3



**2.**



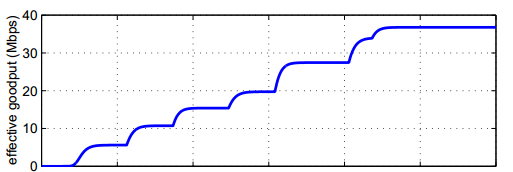
When d=d0=1, we can get Kdb = -10-23 = -33dbm



So 

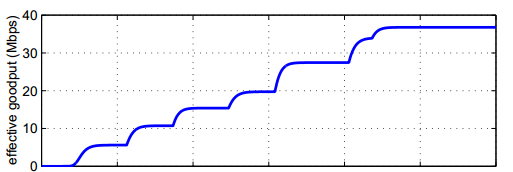
The following pictures are the goodput of 802.11a as a function of log scale distance for path loss exponents 2 and 4.

η= 2



40 37.5 35 32.5 30 27.5 25

η= 4



20 18.7 17.5 16.25 15 13.75 12.5

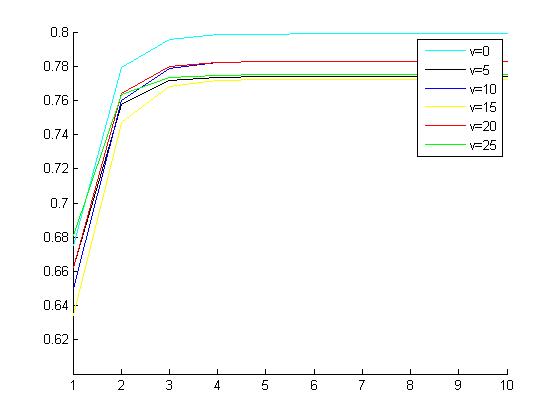
**3.** Due to the threshold size of pow(3), I set the k to 3 in the raygen.m. Based on the mobile speed from [0; 5; 10; 15; 20; 25], run the modified raygen.m and compare the values in power\_ray series with the threshold. If the values above the threshold, then we say which are in good state(1), otherwise in bad state(0). Then I use 6 arrays state[] to hold the results state0, state5, state10, state15, state20, state25. In addition, count the number of transitions from good to bad, Count10, and the number of transitions from bad to good, Count01. Then the parameters P01 and P10 can be calculated using the following equations:

The values of P01 and P10 over different mobile speed are listed in the following table.

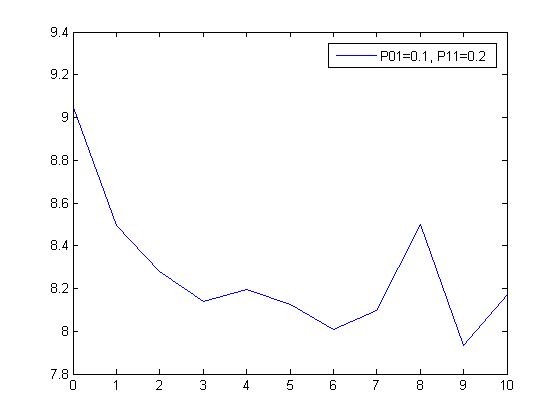
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Speed | 0 | 5 | 10 | 15 | 20 | 25 |
| P01 | 0.6741 | 0.6615 | 0.6490 | 0.6330 | 0.6613 | 0.6800 |
| P10 | 0.1697 | 0.1933 | 0.1801 | 0.1865 | 0.1834 | 0.1975 |

The transition matrix is . So the probability that the channel is in good state in k discrete time steps, given that currently in bad state is PK’s P01. The following is the figure.

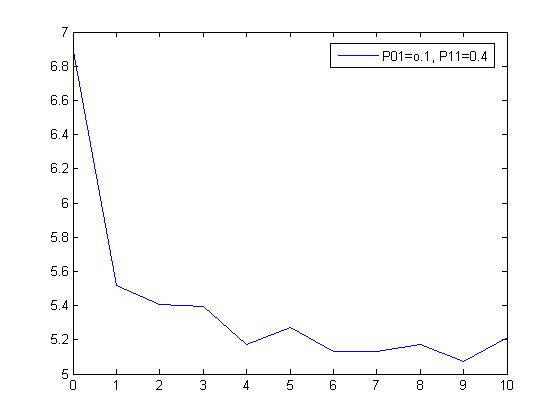


**4.** My policy is: transmit a packet and see whether this packet is successful. If it is successful, then transmit at the next slot immediately. Otherwise, wait for K time slots and then transmit again. Run this policy over a long time period to find at which K steps, we can minimizes ETX - W\*Throughput.

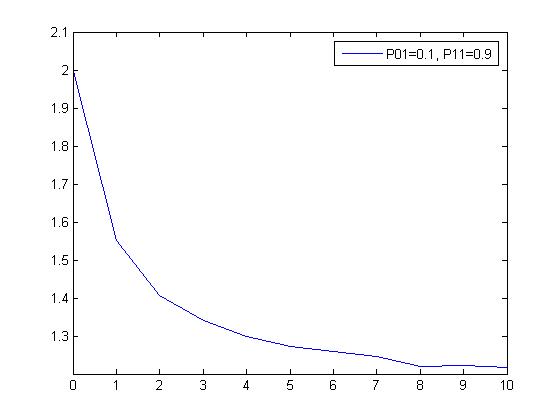
P01 = 0.1 and P11 = 0.2, K = 9



P01 = 0.1 and P11 = 0.4, K = 9



P01 = 0.1 and P11 = 0.9, K = 8



5.

1.

**6.**

(1).

It is difficult to retrieve the original signal after using the ZF technique because ZF also amplify the noise. In order to take account for the noise we have to incorporate it in our solution, so that not to amplify the noise. If we were working in discrete time, then zero forcing will try to null the effect of the channel or make it zero at these points while MSE will try to minimize the mean squared error to avoid channel nulls. Moreover, the MMSE-LE always performs better than the ZFE and is the same complexity of implementation. Nevertheless, it is slightly more complicated to describe and analyze than is the ZFE.

(2).