Monte-Carlo Simulation for generating ROC curves - aksensi

$$P_d = erfc_*\left(\frac{\log \eta}{d} + \frac{d}{2}\right) \quad \text{and} \quad P_f = erfc_*\left(\frac{\log \eta}{d} - \frac{d}{2}\right)$$

Using *normcdf* function of MATLAB for simulation of the ROC curve for 3 values of $d = \frac{m}{\sigma}$ i.e, 0.5, 1, 2; as given in the question. The syntax of *normcdf(Thresold, mean, standard deviation)* and for our case the mean and variance is 0 and 1 respectively. The threshold for P_d is $\frac{\log \eta}{d} + \frac{d}{2}$ and for P_f is $\frac{\log \eta}{d} - \frac{d}{2}$.

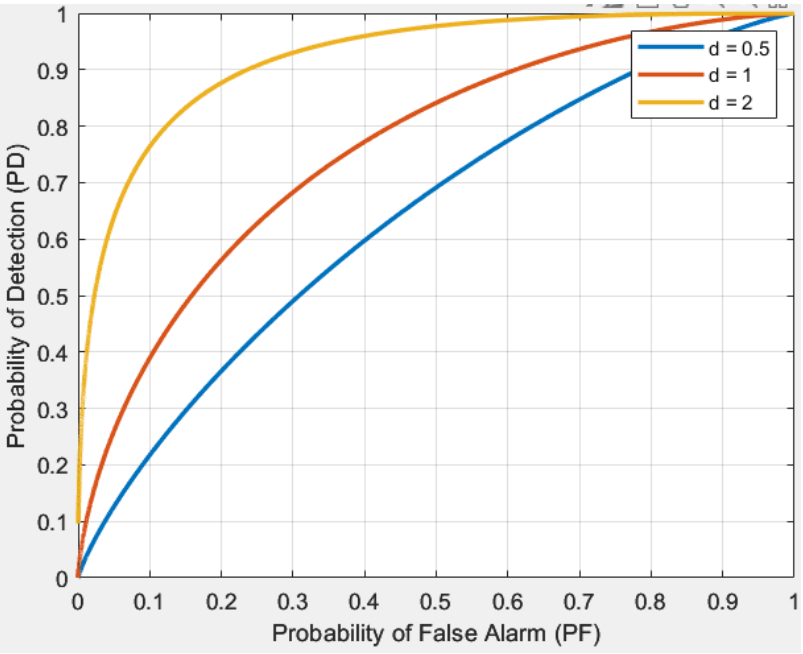
The code for the MATLAB Theoritical simulation is:

```
for d = [0.5 1 2]
   Pf = 1 - normcdf(log(eta)/d + d/2,0,1);
   Pd = 1 - normcdf(log(eta)/d - d/2,0,1);
   plot(Pf,Pd,'LineWidth',2)
   hold on
end
```

Here, the value of eta or η ranges from 0 to ∞ as it cannot be a negative value because log of a negative value is undefined. MATLAB does not support ∞, therefore, I have considered η to be a value from 0 to 100 with an increment of 0.01. The line below describes the statement.

```
eta = 0:0.01:100;
```

The ROC results for the Theoritical Simulation is given below:

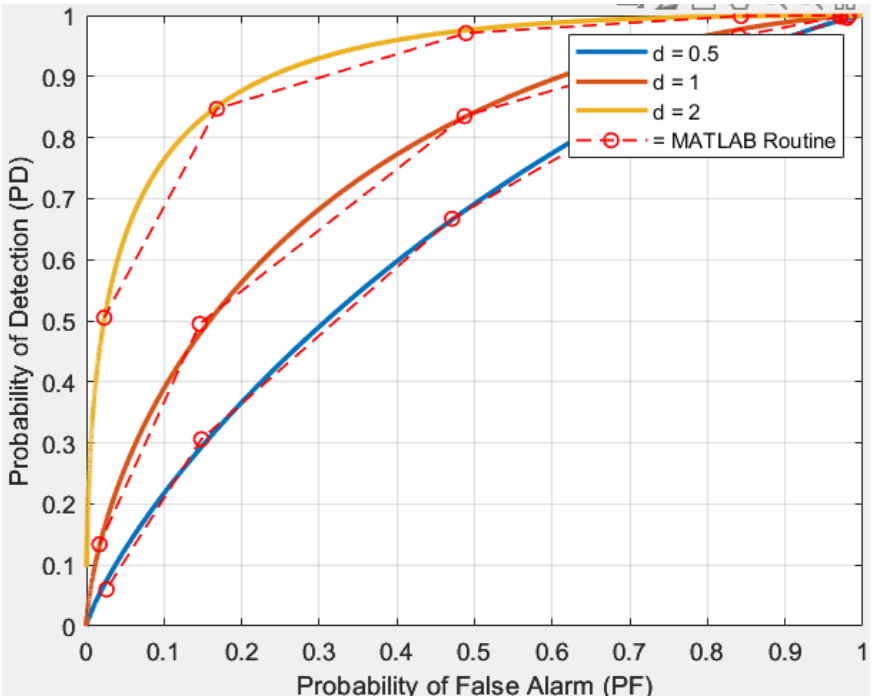


For Monte - Carlo SImultaion, the seed considered for my first and last name is (16+16+16+16) as Akshay Khanna has 4 a's. The thresholds considered for simulation is [-2,-1,0,1,2] and for this I am using *linspace()*.

The entire code for the MATLAB Monte - Carlo simulation and Theoritical SImulation is:

```
clc;
close all;
clear all;
seed = (16+16+16+16);
rng(seed, 'twister');
N = 1000;
mSigma = [0.5, 1, 2];
SNR = [-6, 0, 6];
sigma = 1;
mean = 0;
thresholds =  linspace(-2,2,5);
PD = zeros(length(mSigma),length(thresholds));
PF = zeros(length(mSigma),length(thresholds));
for k = 1:length(mSigma)
   for i = 1:N
       for j = 1:length(thresholds)
           n = normrnd(mean, sigma);
           m = mSigma(k)*sigma;
           threshold = thresholds(j);
           recievedSignal = m+n;
           H_0 = n;
           H_1 = recievedSignal;
           PD(k, j) = PD(k, j) + (H_1 >= threshold);
           PF(k, j) = PF(k, j) + (H_0 >= threshold);
       end
   end
end
PD = PD/N;
PF = PF/N;
eta = 0:0.01:100;
% Plot the ROC curve for Theoretical routine
for d = [0.5 1 2]
   Pf = 1 - normcdf(log(eta)/d + d/2,0,1);
   Pd = 1 - normcdf(log(eta)/d - d/2,0,1);
   plot(Pf,Pd,'LineWidth',2)
   hold on
   grid on
end
hold on
% Plot the ROC curve for MATLAB routine
for k = 1:length(mSigma)
   plot(PF(k,:), PD(k,:),'r--o','LineWidth',1);
   hold on;
end
xlabel('Probability of False Alarm (PF)');
ylabel('Probability of Detection (PD)');
legend('d = 0.5','d = 1','d = 2', '= MATLAB Routine');
```

The ROC comparison results between Theoritical Simulaltion and Monet - Carlo Simulationis given below:


d = 0.5
d = 1
d = 2
= MATLAB Routine
Probability of Detection (PD)
Probability of False Alarm (PF)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1