Contents


```
clear all; close all; clc;
```

Lab 2

1 - Context and Main assumptions

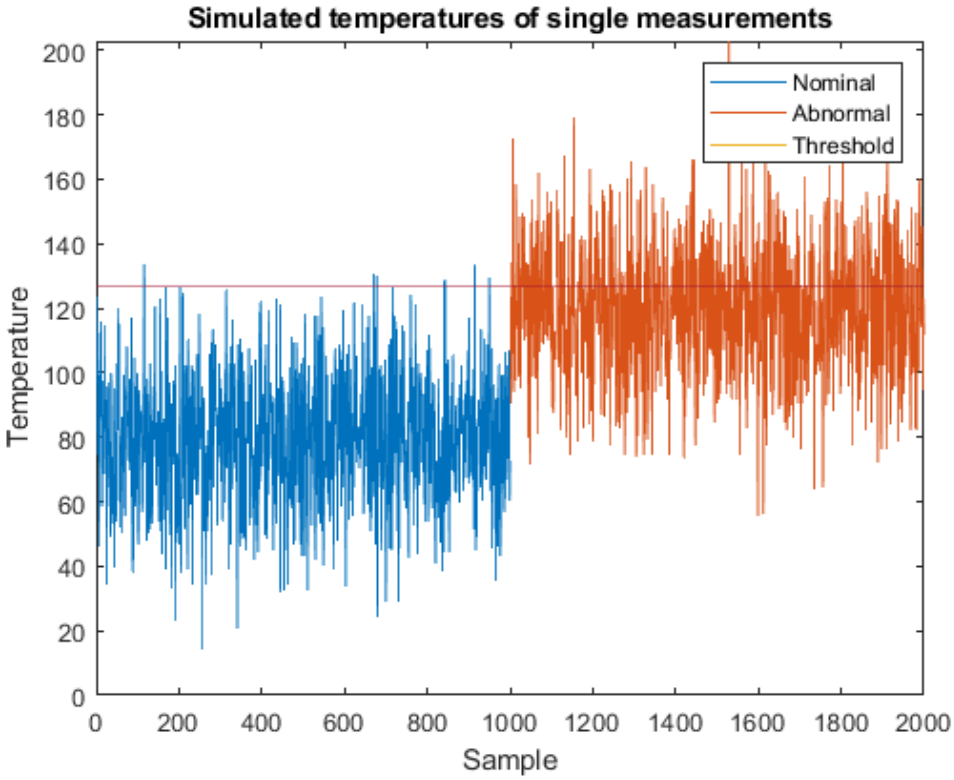
```
% Initial data points
T0 = 80;    % nominal bearing temperature
T1 = 120;   % abnormal bearing temperature
sigma_T = sqrt(400);    % variance/noise of temperature measurement
gamma_pf = 0.01;   % false probability of 1%
```

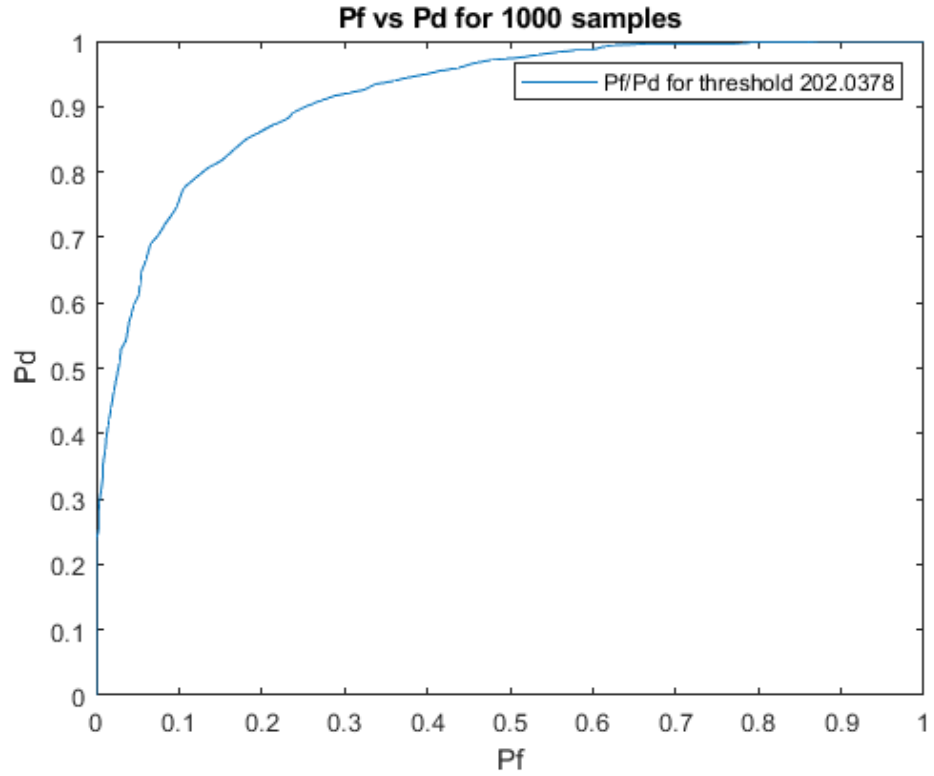
2 - Use of a single measurement sample

```
H0_prime = sigma_T * sqrt(2)*erfinv(1 - 2 * (gamma_pf)) + T0;
threshold = H0_prime;
disp('Threshold value:');
disp(threshold);

Pd = 0.5 * (1 - erf(((threshold - T1) / sigma_T)) / sqrt(2));
disp('Probability of detection:');
disp(Pd * 100);

% Experimentally validate data
k = 1000;   % number of samples
k_T0 = T0 + sigma_T * randn(1, k);  % create k nominal samples
k_T1 = T1 + sigma_T * randn(1, k);  % create k abnormal samples
figure
plot((1:k), k_T0, (k + 1:2*k), k_T1, (1:2*k), threshold*ones(2*k));
xlabel('Sample')
ylabel('Temperature')
title(['Simulated temperatures of single measurements'])
legend('Nominal', 'Abnormal', 'Threshold')

k_T0_above_threshold = sum(k_T0 > threshold);   % number of nominal samples above threshold, false detection
false_detection_ratio = k_T0_above_threshold / k;
disp('False detection percent:');
disp(false_detection_ratio * 100);

k_T1_below_threshold = sum(k_T1 > threshold);   % number of abnormal samples above threshold, faults detected
detected_faults_ratio = k_T1_below_threshold / k;
disp('Fault detection percent:');
disp(detected_faults_ratio * 100);

% Plotting ROC curve
Pfs = [];
Pds = [];
for threshold = (min(k_T0) : max(k_T1))
    k_T0_above_threshold = sum(k_T0 > threshold);   % number of nominal samples above threshold, false detection
    false_detection_ratio = k_T0_above_threshold / k;
    Pfs(end + 1) = false_detection_ratio;
```

```
    k_T1_below_threshold = sum(k_T1 > threshold);   % number of abnormal samples above threshold, faults detected
    detected_faults_ratio = k_T1_below_threshold / k;
    Pds(end + 1) = detected_faults_ratio;
end
figure
plot(Pfs, Pds)
title(['Pf vs Pd for ', num2str(k), ' samples'])
xlabel('Pf')
ylabel('Pd')
legend(['Pf/Pd for threshold ', num2str(threshold)]);
```

```
Threshold value:
  126.5270

Probability of detection:
   37.4284

False detection percent:
    0.9000

Fault detection percent:
   36.4000
```



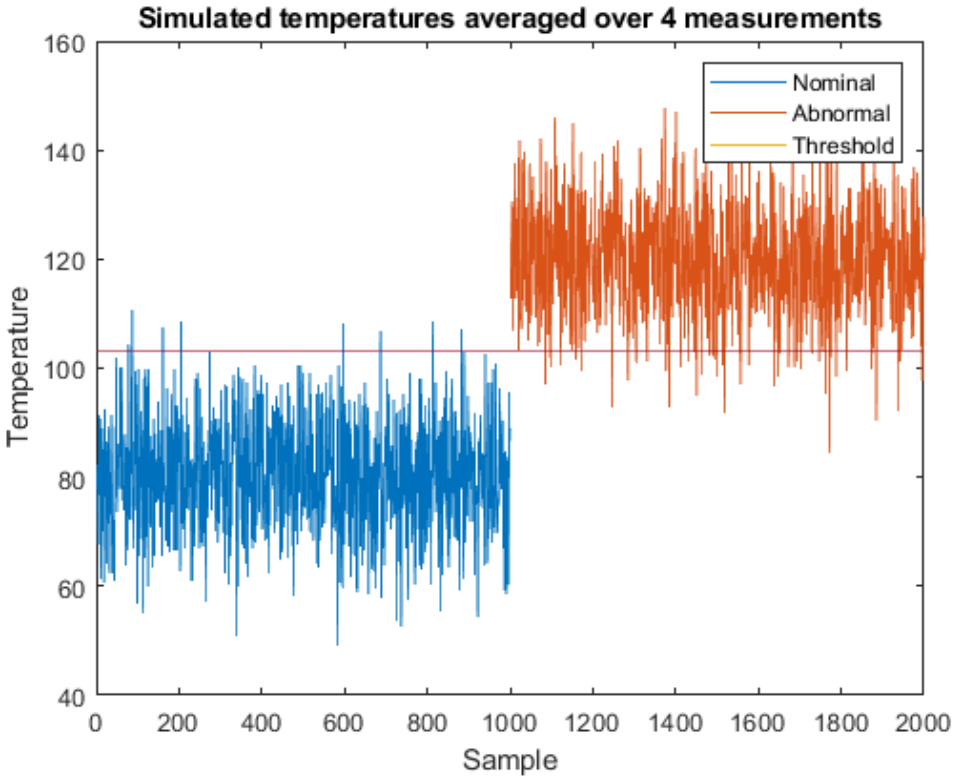


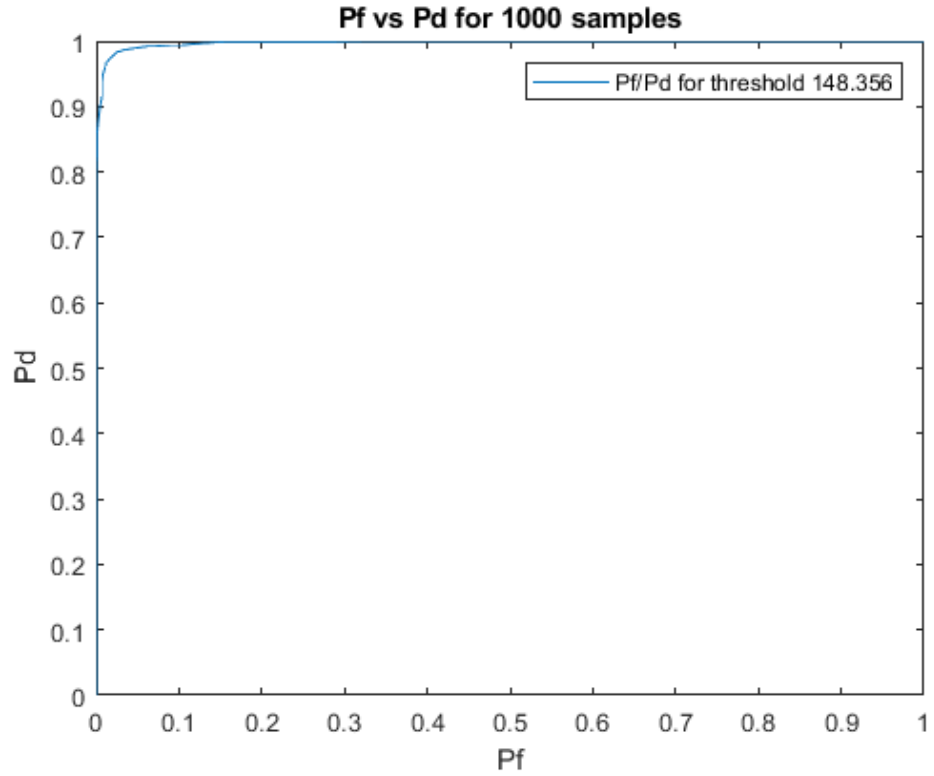
3 - Use of N measurement samples

```
% Calculate number of samples needed to achieve desired properties
Pf_desired = 0.01;
Pd_desired = 0.9;
% N = [(sigma * (Q_inv(Pf_desired) - Q_inv(Pd_desired)) / (T1 - T2)] ^ 2
% N = [(sigma * (Q_inv(Pd_desired) + T1) / (sigma * Q_inv(Pd_desired)) / (T1 - T2)] ^ 2
N = ceil((sigma_T * (sqrt(2)*erfinv(1-2*(Pf_desired)) - sqrt(2)*erfinv(1-2*(Pd_desired))) / (T1 - T0)) ^ 2);
```

```
disp('Number of samples needed:');
disp(N);

% Calculate matching threshold
threshold = (sigma_T / sqrt(N)) * (sqrt(2)*erfinv(1-2*(Pf_desired))) + T0;
disp('Threshold for N samples');
disp(threshold);

% Experimentally validate N
k_T0 = zeros(1, k);
k_T1 = zeros(1, k);
for n = 1 : k
    k_T0(n) = mean(T0 + sigma_T * randn(1, N));    % create k nominal samples, averaged over N measurements
    k_T1(n) = mean(T1 + sigma_T * randn(1, N)); % create k abnormal samples, averaged over N measurements
end
figure
plot((1:k), k_T0, (k + 1:2*k), k_T1, (1:2*k), threshold*ones(2*k));
xlabel('Sample')
ylabel('Temperature')
title(['Simulated temperatures averaged over ', num2str(N), ' measurements'])
legend('Nominal', 'Abnormal', 'Threshold')

k_T0_above_threshold = sum(k_T0 > threshold);   % number of nominal samples above threshold, false detection
false_detection_ratio = k_T0_above_threshold / k;
disp('False detection percent:');
disp(false_detection_ratio * 100);

k_T1_below_threshold = sum(k_T1 > threshold);   % number of abnormal samples above threshold, faults detected
detected_faults_ratio = k_T1_below_threshold / k;
disp('Fault detection percent:');
disp(detected_faults_ratio * 100);

% Plotting ROC curve
Pfs = [];
Pds = [];
for threshold = (min(k_T0) : max(k_T1))
    k_T0_above_threshold = sum(k_T0 > threshold);   % number of nominal samples above threshold, false detection
    false_detection_ratio = k_T0_above_threshold / k;
    Pfs(end + 1) = false_detection_ratio;
    k_T1_below_threshold = sum(k_T1 > threshold);   % number of abnormal samples above threshold, faults detected
    detected_faults_ratio = k_T1_below_threshold / k;
    Pds(end + 1) = detected_faults_ratio;
end
figure
plot(Pfs, Pds)
title(['Pf vs Pd for ', num2str(k), ' samples'])
xlabel('Pf')
ylabel('Pd')
legend(['Pf/Pd for threshold ', num2str(threshold)]);
```

```
Number of samples needed:
     4

Threshold for N samples
  103.2635

False detection percent:
    0.8000

Fault detection percent:
   95.1000
```





Published with MATLAB® R2018b