# University of Oulu

Faculty of Information Technology and Electrical Engineering - ITEE


## Statistical Signal Processing 1

---

# MATLAB Project

---


**Group:** 14

**Students:**

- AZZAZ Aissa (Number: 2207335)
- BOULFRAD Mourad (Number: 2207592)


**Due date**: November 24th 2022

## Detection of a known signal

The receiver observes the signal

$$x[n] = As[n] + w[n], \quad n = 0, 1, \dots N-1$$

where both $A$ and $S = [s[0], s[1], s[2], \dots s[N-1]]^T$ are deterministic variables and known to the receiver. $S$ represents a unit power DC level and $w[n]$ is a white Gaussian noise such that $w[n] \sim \mathcal{N}(0, \sigma^2)$, thus

$$\text{Power}\{S\} = \frac{1}{N} S^T S = \frac{1}{N} \begin{bmatrix} \alpha \\ \alpha \\ \vdots \\ \alpha \end{bmatrix} \begin{bmatrix} \alpha & \alpha & \dots & \alpha \end{bmatrix} = \frac{1}{N} \sum_{n=0}^{n=N-1} \alpha^2 = \frac{\alpha^2 N}{N} \alpha^2 = 1 \implies \alpha = 1 \implies S = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

The receiver aims to detect whether the signal $AS$ is present or not. This can be formulated using the two hypotheses

$$\begin{cases} H_0 : & X = W \\ H_1 : & X = AS + W \end{cases}$$

## 1. Deriving a detector for the signal $AS$

To avoid confusion, the signal is $AS$ is denoted as

$$D = AS$$

Because the variable $A$ and $S$ deterministic variables and known to the receiver, we can use Neyman-Pearson Theorem to derive a detector for the signal. The samples are independent, and the noise is WGN, so the likelihood ratio is

$$L(X) = \frac{p(X; H_1)}{p(X; H_0)} > \gamma$$

$$L(X) = \frac{\Pi_{n=0}^{N-1} p(X; H_1)}{\Pi_{n=0}^{N-1} p(X; H_0)} = \frac{\Pi_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2\sigma^2}(x[n]-As(n))^2}}{\Pi_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2\sigma^2}(x[n]-0)^2}} = \frac{\frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} e^{\frac{-1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n]-As(n))^2}}{\frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} e^{\frac{-1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n])^2}} = \frac{e^{\frac{-1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n]-As(n))^2}}{e^{\frac{-1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n])^2}}$$

Using matrix form, we get

$$L(X) = \frac{e^{\frac{-1}{2\sigma^2}(X-AS)^T(X-AS)}}{e^{\frac{-1}{2\sigma^2}X^T X}} = \frac{e^{\frac{-1}{2\sigma^2}(X-D)^T(X-D)}}{e^{\frac{-1}{2\sigma^2}X^T X}} = e^{\frac{-1}{2\sigma^2}\left((X-D)^T(X-D)-X^T X\right)} > \gamma$$

Applying the $\log$ to both sides, we get

$$\log\left(L(X)\right) = \frac{-1}{2\sigma^2}\left((X-D)^T(X-D) - X^T X\right) > \log(\gamma)$$

$$\log\left(L(X)\right) = \frac{-1}{2\sigma^2}\left(X^T X - X^T D - D^T X + D^T D - X^T X\right) = \frac{-1}{2\sigma^2}\left(-X^T D - D^T X + D^T D\right) > \log(\gamma)$$

Since $D$ and $X$ are column vector we have $X^T D = D^T X = (scaler)$, we get

$$\log\left(L(X)\right) = \frac{-1}{2\sigma^2}\left(-2D^T X + D^T D\right) > \log(\gamma)$$

$$T(X) : D^T X > \sigma^2 \log(\gamma) + \frac{1}{2} D^T D = \gamma'$$

$$T(X) : D^T X > \gamma'$$

The test statistics $T(X)$ is matched filter that is matched to the signal $D = AS$. The energy of the signal and the energy to noise ratio are

$$E_D = D^T D = \sum_{n=0}^{N-1}(As(n))^2 = A^2 \sum_{n=0}^{N-1}(1)^2 = A^2 N \; ; \; ENR = \frac{E_D}{\sigma^2} = \frac{NA^2}{\sigma^2}$$

Using the results of the matched filter, we get the pdf of the test statistics, the probabilities of false alam $P_{FA}$ and detection $P_D$ as

$$\begin{cases} \text{Under } H_0: & T(X) \sim \mathcal{N}(0, E_D\sigma^2) \\ \text{Under } H_1: & T(X) \sim \mathcal{N}(E_D, E_D\sigma^2) \end{cases}$$

$$.i.e \begin{cases} \text{Under } H_0: & p(T(X); H_0) = \frac{1}{\sqrt{2\pi E_D\sigma^2}} e^{\frac{-1}{2E_D\sigma^2} T^2(X)} \\ \text{Under } H_1: & p(T(X); H_1) = \frac{1}{\sqrt{2\pi E_D\sigma^2}} e^{\frac{-1}{2E_D\sigma^2} (T(X)-E_D)^2} \end{cases}$$

$$P_{FA} = p(H_1|H_0) = \mathbf{Q}\left(\frac{\gamma'}{\sqrt{\sigma^2 E_D}}\right) = \mathbf{Q}\left(\frac{\gamma'}{\sqrt{\sigma^2 A^2 N}}\right)$$

$$P_D = p(H_1|H_1) = \mathbf{Q}\left(\mathbf{Q}^{-1}\left(P_{FA}\right) - \sqrt{ENR}\right) = \mathbf{Q}\left(\mathbf{Q}^{-1}\left(P_{FA}\right) - \sqrt{\frac{NA^2}{\sigma^2}}\right)$$

## 2. Implementing a Monte Carlo simulation for the detector in MATLAB, and plotting and comparing the theoretical and the simulated PDFs

The code below, found in *Task1_1.m* file, starts by setting the simulation parameters (N, MC, A, $\sigma^2$) and then generates the signal $D = AS$ and the noise $W$ (the code is vectorized to avoid slow for loops). After that, the captured signal $X$ is generated for both hypotheses. Then, the decision statistic $T(X)$ are calculated for both hypothesis. The code then generates Figure 1 the simulated and theoretical PDFs of the decision statistic for both hypotheses. The simulated PDFs perfectly match the theoretical ones derived from the matched filter approach.

```
close all ; clear ; clc ;
rng(0) ; % reset the random number generator (for reproducibility)
% format long eng

%%%%%%%% Q2

MC = 100000 ; % number of Monte Carlo loops
A = sqrt(20) ;
sigma_squared = 2 ; % Noise variance
N = 100 ; % Number of samples in the data

S = ones(N,1) ;

%%% Generating of the samples
noise = randn(N,MC)*sqrt(sigma_squared) ; % W ==> N columns of WGN repeated MC rows
D = A*S ; % the signal to be detected

X_no_signal = noise ;
X_with_signal = D + noise ;

%%% Calculating the test statistic
T_H0 = D'*X_no_signal ; % test statistic  under H0
T_H1 = D'*X_with_signal ; %test statistic under H1

B0 = histogram(T_H0,'Normalization','pdf','DisplayStyle','stairs','LineWidth',1.5);
hold on
B1 = histogram(T_H1,'Normalization','pdf','DisplayStyle','stairs','LineWidth',1.5);

E_D = D'*D ; % energy of D=A*S

% pdf ranges on x axis ie T(X) range
x_H0 = linspace(B0.BinLimits(1),B0.BinLimits(2),100) ;
x_H1 = linspace(B1.BinLimits(1),B1.BinLimits(2),100) ;
```

```
T_H0_theory = normpdf(x_H0,0, sqrt(E_D*sigma_squared)) ;
T_H1_theory = normpdf(x_H1,E_D, sqrt(E_D*sigma_squared)) ;

plot(x_H0,T_H0_theory,'gx--',x_H1,T_H1_theory,'k*--')

grid minor
legend('Simulated p(T(X);H_{0})','Simulated p(T(X);H_{1})', ...
         'Theoretical p(T(X);H_{0})','Theoretical p(T(X);H_{1})',Location='north')
xlabel('Test statistic'); ylabel('PDF')
hold off
```

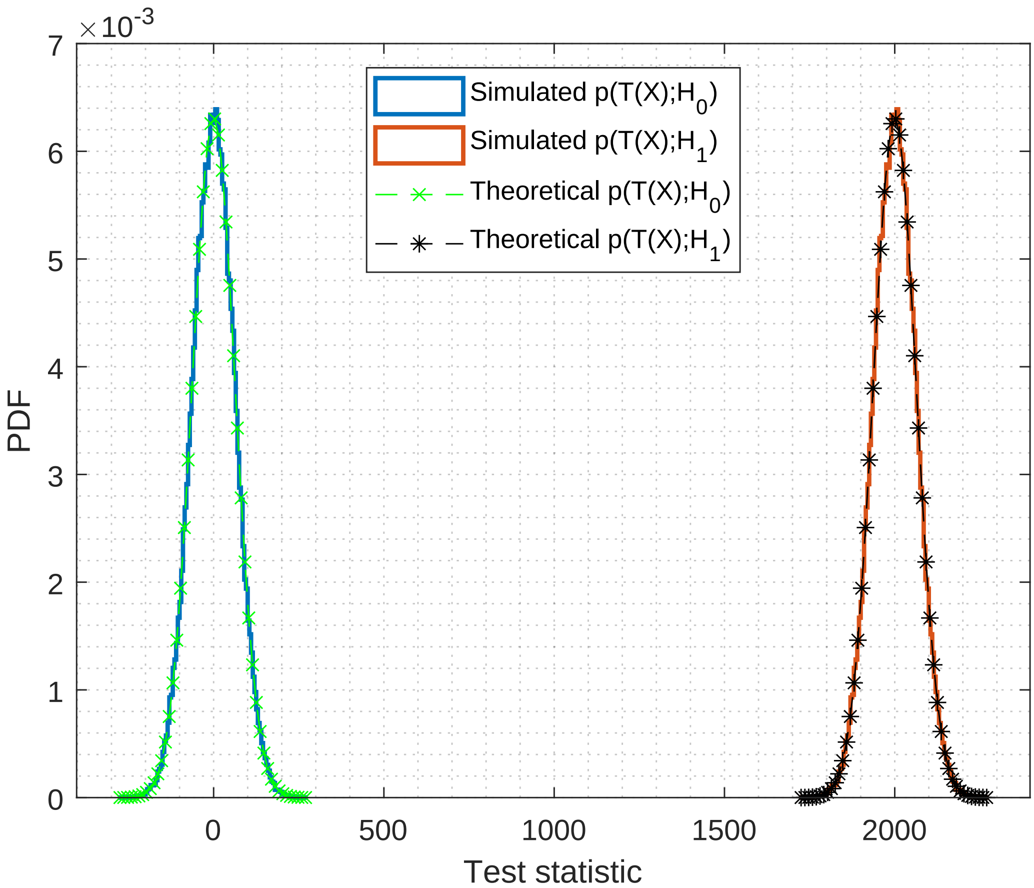


**Figure 1:** Simulated and theoretical PDFs of the decision statistics T(X) under $H_0$ and $H_1$

## 3. Plot the ROC curve of the detector

The receiver operating characteristics (shortly ROC) plot is a curve that shows the relation between $P_D$ and $P_{FA}$ for a certain ENR level. Because we have the ENR level (set by the given parameters) and the expression of $P_D$ in terms $P_{FA}$, we can plot the ROC curve.

The code below, found in *Task1_1.m* file, calculates the ENR and generates values of the threshold $\gamma$ and calculates $P_{FA}$ and $P_D$ for them, and then it plots the ROC curve. Figure 2 shows the ROC curve. More than 1000 equispaced value of $\gamma'$ are used to obtain a better insight on the ROC at low $P_{FA}$. It's clear that for the given parameters ($\sigma^2$,$A$,$N$), the detector is highly likely not make any false alarms as $P_D$ converges quickly (starting from $P_{FA} = 10^{-200}$) to 1 with small value of $P_{FA}$ and this is also reflected in the separation between the PDFs of the test statistic, Figure 1, which allows for a more correct decision and a relaxed margin of error to play with and a relaxed threshold.

```
%%%%%%%% Q3
```

```
ENR = E_D/sigma_squared ; % Energy to noise Ratio

gamma = -MC/10:0.01:+MC/10 ;
P_FA = qfunc(gamma./sqrt(sigma_squared*E_D)) ;
P_D = qfunc(qfuncinv(P_FA)-sqrt(ENR)) ;

figure
loglog(P_FA,P_D)

grid minor
xlabel('P_{FA}'); ylabel('P_{D}')
```

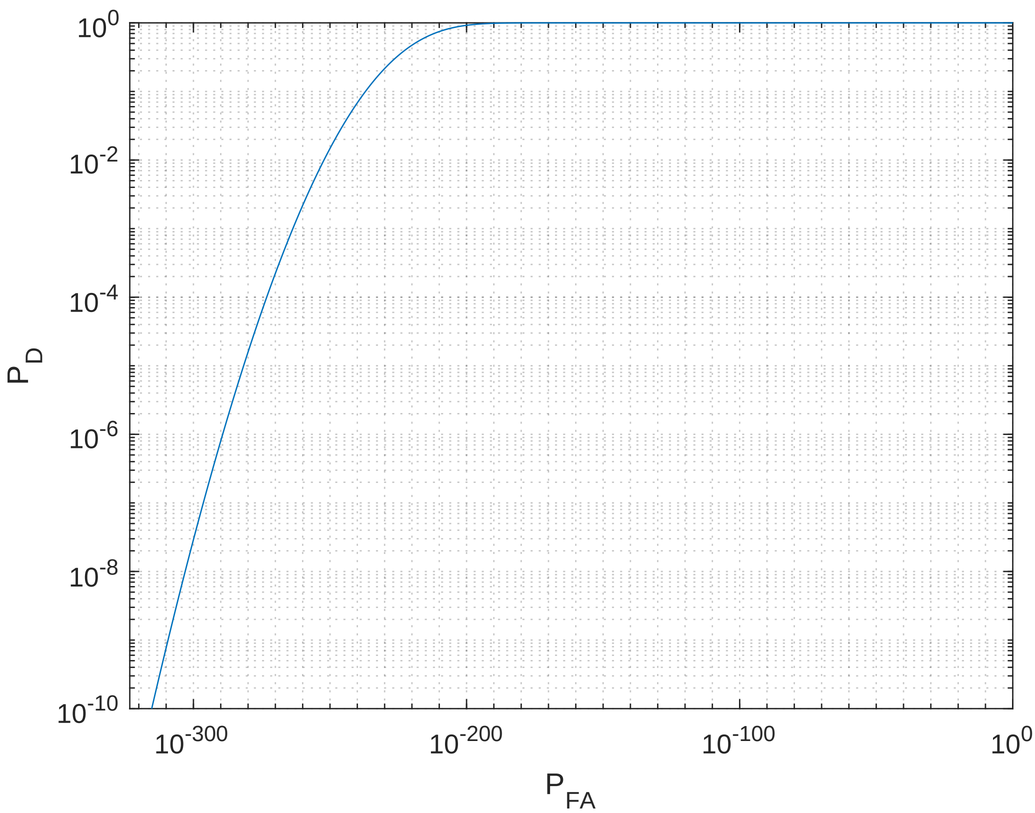


**Figure 2:** The ROC curve with the given parameters

## 4. Changing $A$ values to plot $P_D$ versus $P_{FA}$ for the given SNR levels

Using the equation of the ENR (SNR) of the first question, we get

$$SNR = ENR = \frac{E_D}{\sigma^2} = \frac{NA^2}{\sigma^2} \implies A = \frac{\sigma^2}{N} SNR$$

The code below, found in *Task1_2.m* file, calculates the values of $A$ for the given SNR values and also calculates corresponding energy of the signal. Then it generates a finite range of the threshold $\gamma'$ which uses it to calculate $P_{FA}$ and $P_D$ for the various SNR levels, and finally plots the results presented in Figure 3. It is clear that no SND value causes the ROC curve to be under $P_D = P_{FA}$ line and this intuitive because when there is no false alarm without a detection involved in it. Also, as the SNR level increases, $P_D$ gets better for a fixed $P_{FA}$ and vice versa.

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

```
MC = 100000 ; % number of Monte Carlo loops
sigma_squared = 2 ; % Noise variance
N = 100 ; % Number of samples in the data

SNR_dB = [-5 0 5 10 15 20] ;
SNR = 10.^(SNR_dB/10) ;
A = sigma_squared*SNR/N ;
E_D = N*A.^2 ; % energy of D=A*S

figure
hold on

gamma = (-MC:0.01:+MC) ;
for ii = 1:length(SNR)
    P_FA = qfunc(gamma./sqrt(sigma_squared*E_D(ii))) ;
    P_D = qfunc(qfuncinv(P_FA)-sqrt(SNR(ii))) ;

    loglog(P_FA,P_D)

    myLegend{ii} = strcat('SNR =', num2str(SNR_dB(ii)), ' dB') ;
end

x_limit = 0:0.01:1 ;
plot(x_limit,x_limit,'--')
myLegend{end+1} = 'P_{D}=P_{FA}' ;
legend(myLegend,'Location','best')
grid minor
xlabel('P_{FA}'); ylabel('P_{D}')
hold off
```

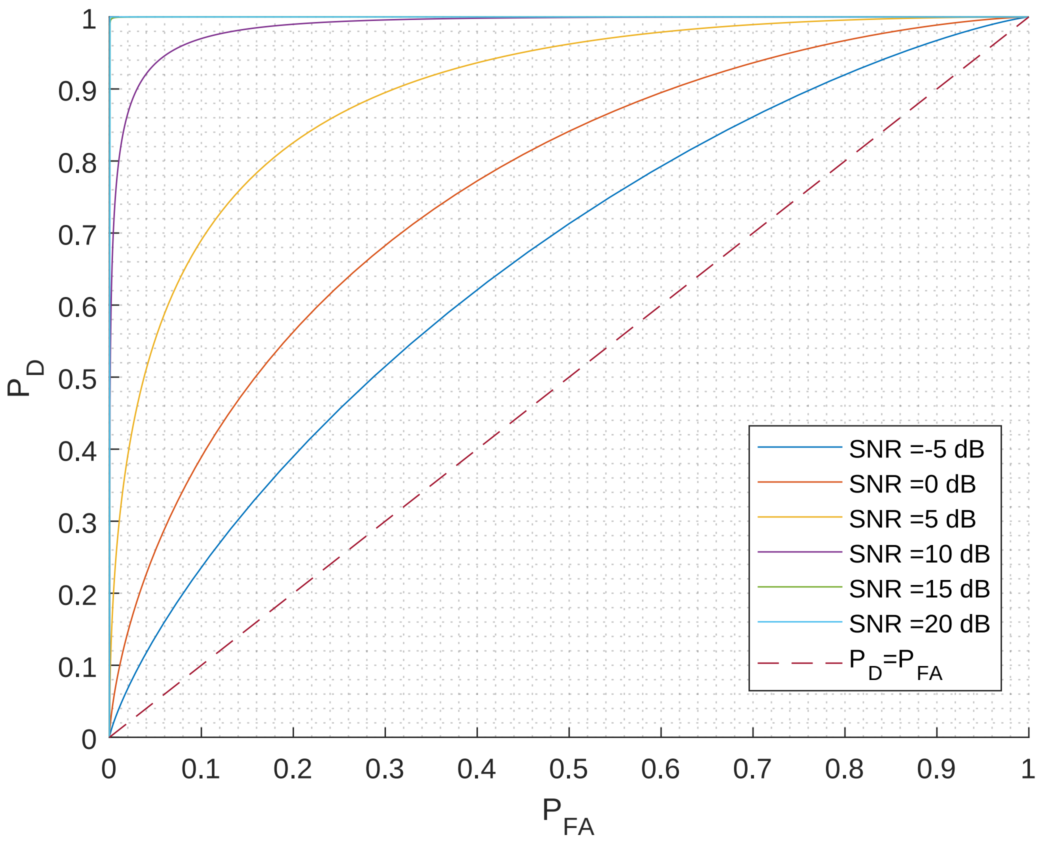


**Figure 3:** The ROC curves for the given SNR levels

## 5. The usage of the detector with unknown $A$

When $S$ is still deterministic and known to the receiver but the value of $A$ is unknown to it, the detector (matched filter) used can not be used as the detector need the value of $A$ to evaluate the test statistics (to generate the matched signal $D^T$).

A simple technique to keep using the same detector is before detection, the receiver estimates $A$. The code below, found in *Task1_3.m* file, examines this technique using $A$ estimator based on the sample mean and the used hypothesis and then performs the same steps to detect the signal as discussed in the second section for the *Task1_1.m* file. Figure 4 shows the simulated PDF of the test statistics based on the estimation and the theoretical PDF based on the true value of $A$. It's clear that still without knowing $A$, the receiver can make a good detection (at least with value given at first for $\sigma^2$, $A$, $N$) since the distribution of the two hypothesis are quite separated.

```
close all ; clear ; clc ;
rng(0) ; % reset the random number generator (for reproducibility)

MC = 100000 ; % number of Monte Carlo loops
A = sqrt(20) ;
sigma_squared = 2 ; % Noise variance
N = 100 ; % Number of samples in the data

S = ones(N,1) ;

%%% Generating of the samples
noise = randn(N,MC)*sqrt(sigma_squared) ; % W ==> N columns of WGN repeated MC rows
D = A*S ; % the signal to be detected
% var(noise,1)
X_no_signal = noise ;
X_with_signal = D + noise ;

%%% Estimating A
A_estimated_H0 = mean(X_no_signal,'all') ;
A_estimated_H1 = mean(X_with_signal,'all') ;

%%% Calculating the test statistics
T_H0 = A_estimated_H0*S'*X_no_signal ; % test statistics when there's no signal (i.e H0)
T_H1 = A_estimated_H1*S'*X_with_signal ; %test statistics when there's a signal (i.e H1)

B0 = histogram(T_H0,'Normalization','probability','DisplayStyle',"stairs");
hold on
B1 = histogram(T_H1,'Normalization','probability','DisplayStyle',"stairs");

E_D = N*A.^2 ; % energy of D=A*S

% pdf ranges on x axis ie T(X) range
x_H0 = linspace(B0.BinLimits(1),B0.BinLimits(2),50) ;
x_H1 = linspace(B1.BinLimits(1),B1.BinLimits(2),50) ;

T_H0_theory = normpdf(x_H0,0, sqrt(E_D*sigma_squared)) ;
T_H1_theory = normpdf(x_H1,E_D, sqrt(E_D*sigma_squared)) ;

plot(x_H0,T_H0_theory,'gx--',x_H1,T_H1_theory,'k*--')

grid minor
legend('Simulated p(T(X);H_{0})','Simulated p(T(X);H_{1})', ...
        'Theoretical p(T(X);H_{0})','Theoretical p(T(X);H_{1})',Location='north')
xlabel('Test statistic'); ylabel('PDF')
hold off
```

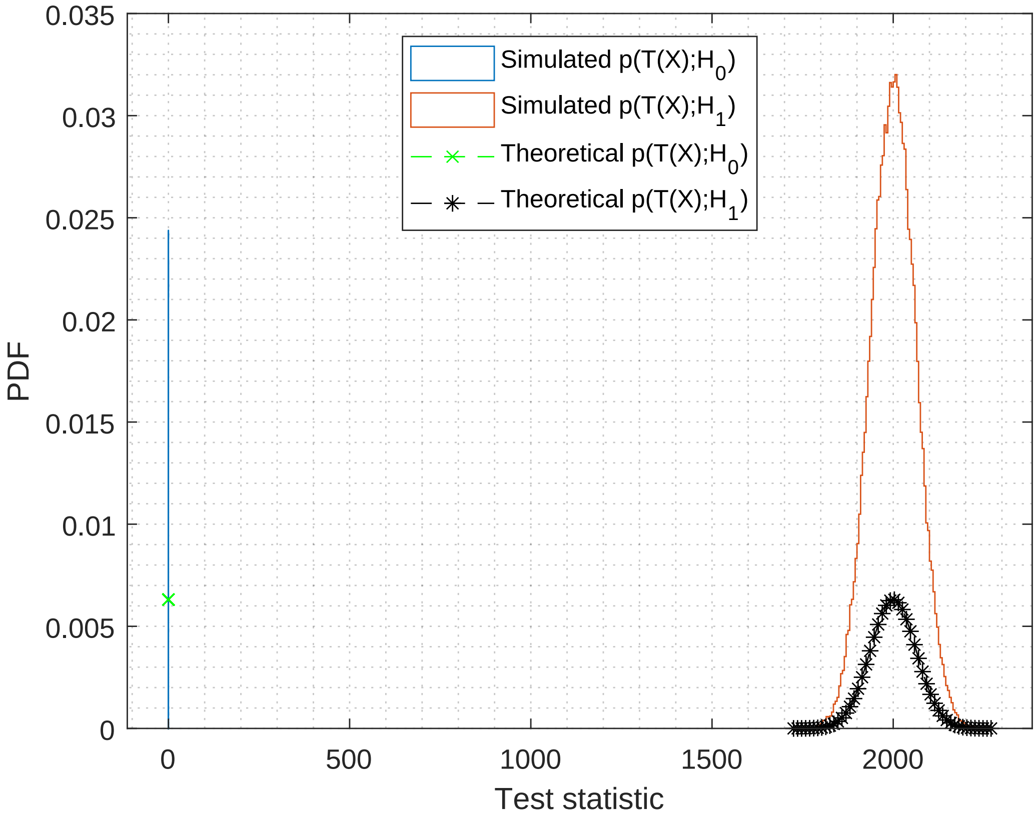


**Figure 4:** Simulated and theoretical PDFs of the decision statistics T(X) with estimated $A$