A Case Study of Basic Sequential Detection Methods

Arindam Bose

***Abstract*—Sequential detection methods have their own advantages over fixed length detection methods. While in fixed length detection methods the performance of the detector is improved by increasing the signal-to-noise ratio which involves increase in the number of signal samples, it also requires increasing time and resources to reach a decision. In sequential design rule, signal samples are processed sequentially to produce a decision statistic which is compared to thresholds and thus we can reduce the average number of samples necessary to reach a decision. In this project we will examine the sequential decision rules for several cases.**

***Index Terms*—Likelihood ratio test, Neaman-Pearson test, Sequential Detection.**

I. Introduction

THIS project familiarizes us with the theory of basic sequential detection methods and investigate their advantages and disadvantages relative to fixed length detection methods. The performance of a detector can be improved by increasing the signal-to-noise ratio [1], [2]. In most cases, the noise power is fixed, and in order to improve performance of the detector, we have to increase the signal energy $N\mu^2$, where μ is the amplitude of a transmitted signal. Due to power constraints we cannot increase the magnitude of the transmitted signal (for instance, the case of wireless communications). For this reason, we attain increased signal-to-noise ratio by increasing the number of data samples N. However, this increase in number of data samples leads to increased time and resources. In order to minimize the number of data samples, sequential detection comes into play. In most cases the average number of data samples needed to reach a desired performance is reduced by the use of sequential detection techniques.

In a sequential decision rule, samples are processed sequentially to form a decision statistic which is compared to both an upper and a lower threshold. When the statistic crosses either threshold, processing stops and a decision is declared. Thus, the number of samples needed to reach a decision depends on the values of the samples. Usually, the average number of samples needed to obtain a desired level of performance of the detector is fewer than the number of samples necessary for a fixed sample length detector.

In this project, we will derive sequential decision rules for several problems. We will then evaluate these rules using Monte Carlo computer simulation, characterizing their performance in terms of designed and actual probabilities of detection (P_D) and probabilities of false alarm (P_{FA}) and in terms of expected number of samples needed to reach a decision.

A. Bose is a PhD student of Electrical and Computer Engineering Department, University of Illinois at Chicago, Chicago, IL 60607, USA (email: abose4@uic.edu).

II. Theory of Sequential Detection

We consider a detection problem between null hypothesis $\mathcal{H}_0$ and alternative hypothesis $\mathcal{H}_1$. We assume that the observations $x[0], x[1], \cdots, x[N-1]$ and that the likelihood functions of the observations under the two hypotheses are $p(x[0], x[1], \cdots, x[N-1]; \mathcal{H}_0)$ and $p(x[0], x[1], \cdots, x[N-1]; \mathcal{H}_1)$ [1]. A sequential test requires two thresholds A and B; we show below how these thresholds are computed to achieve desired values of P_D and P_{FA}. Given values of A and B, the sequential decision algorithm begins with $N = 1$ and a single observation $x[0]$, and makes decisions as follows:

- Choose $\mathcal{H}_0$ if

$$L(x[0], x[1], \cdots, x[N-1]) = \frac{p(x[0], x[1], \cdots, x[N-1]; \mathcal{H}_1)}{p(x[0], x[1], \cdots, x[N-1]; \mathcal{H}_0)} \leq B. \quad (1)$$

- Choose $\mathcal{H}_1$ if

$$L(x[0], x[1], \cdots, x[N-1]) = \frac{p(x[0], x[1], \cdots, x[N-1]; \mathcal{H}_1)}{p(x[0], x[1], \cdots, x[N-1]; \mathcal{H}_0)} \geq A. \quad (2)$$

- Otherwise, get another sample, increase N by one, and repeat the test.

Unlike the Neyman-Pearson test, with a sequential test we can specify both P_D and P_{FA} for a detector; the values of P_D and P_{FA} determine the average number of samples that will be needed to reach a decision. The definition of P_D and P_{FA} over the region R_1 are as follows:

$$P_{FA} = \int_{R_1} p(\boldsymbol{x}; \mathcal{H}_1) d\boldsymbol{x}$$

$$P_D = \int_{R_1} p(\boldsymbol{x}; \mathcal{H}_0) d\boldsymbol{x}$$

where $\boldsymbol{x} = [x[0], x[1], \cdots, x[N-1]]$.

We now approximate expressions for A and B in terms of the desired values P_D and P_{FA} as illustrated in Fig. 1. To obtain A, we consider the situation where the test terminates with N samples having chosen $\mathcal{H}_1$. In this case, (2) holds. We assume that the amount by which $L(x[0], x[1], \cdots, x[N-1])$ exceeds A in this test is negligibly small; this assumption is typically valid when a large number of samples is required to reach a decision. Under this assumption,

$$p(x[0], x[1], \cdots, x[N-1]; \mathcal{H}_1) = Ap(x[0], x[1], \cdots, x[N-1]; \mathcal{H}_0)$$

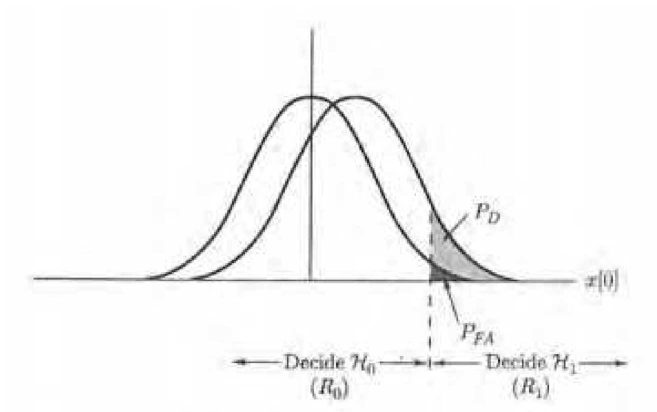

Fig. 1. Decision region and probabilities.

Integrating both sides of this equality over R_1 in Fig. 1, we obtain,

$$P_D = AP_{FA},$$

from which we can solve for A:

$$A = \frac{P_D}{P_{FA}} \tag{3}$$

Similarly, to obtain B, we consider the situation where the test terminates with N samples and $\mathcal{H}_0)$ is chosen. In this case, (1) holds. We assume that the amount by which $L(x[0], x[1], \cdots, x[N-1])$ is smaller than B is negligibly small:

$$\begin{aligned} &p(x[0], x[1], \cdots, x[N-1]; \mathcal{H}_1) \\ &= Bp(x[0], x[1], \cdots, x[N-1]; \mathcal{H}_0) \end{aligned}$$

Integrating both sides of this equality over R_0, we obtain,

$$1 - P_D = B(1 - P_{FA}),$$

from which we can solve for B:

$$B = \frac{1 - P_D}{1 - P_{FA}} \tag{4}$$

III. Problem Description

In this project sequential detection scheme is implemented for three different cases. In each case we deal with two hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$. A brief description of these three cases is as follows.

1) Case I: We will implement a very simple sequential test to become familiar with its operation and simulation. The two hypotheses are

$$\begin{aligned} \mathcal{H}_0 &: \boldsymbol{x}[n] = \boldsymbol{w}[n] \qquad , n = 0, \cdots, N-1 \\ \mathcal{H}_1 &: \boldsymbol{x}[n] = \mu + \boldsymbol{w}[n], n = 0, \cdots, N-1 \end{aligned}$$

where μ is a known positive constant and $\boldsymbol{w}[n]$ is a Gaussian white noise sequence with mean zero and known variance σ^2.

2) Case II: We will consider detection of a sinusoidal signal with known amplitude, frequency, and phase using a sequential detector. This is motivated by problems in radar and active sonar. The detection problem is the following:

$$\begin{aligned} \mathcal{H}_0 &: \boldsymbol{x}[n] = \boldsymbol{w}[n] \qquad\qquad , n = 0, \cdots, N-1 \\ \mathcal{H}_1 &: \boldsymbol{x}[n] = \boldsymbol{s}[n] + \boldsymbol{w}[n], n = 0, \cdots, N-1 \end{aligned}$$

where $\boldsymbol{w}[n]$ is a Gaussian white noise (WGN) sequence with mean zero and known variance σ^2 and $\boldsymbol{s}[n]$ is a sinusoidal signal:

$$\boldsymbol{s}[n] = \mu \cos(\omega n), \qquad -\pi \le \omega \le \pi$$

3) Case III: Here the Hypotheses are as following:

$$\begin{aligned} \mathcal{H}_0 &: \boldsymbol{x}[n] = \boldsymbol{w}[n] \qquad\qquad , n = 0, \cdots, N-1 \\ \mathcal{H}_1 &: \boldsymbol{x}[n] = \boldsymbol{s}[n] + \boldsymbol{w}[n], n = 0, \cdots, N-1 \end{aligned}$$

where $\boldsymbol{w}[n]$ is Gaussian white noise (WGN) with zero mean and known variance σ_N^2 and the signal $\boldsymbol{s}[n]$ is another WGN sequence with zero mean and known variance σ_S^2. This signal model might apply to passive underwater detection of ships, in which the signal from a ship in a given frequency band is due to the interaction of the hull with the water.

IV. Numerical Analysis

In this section we will try to analyze the hypotheses and derive the mathematical definition of the thresholds for each cases described in section II.

1) Case I: The two hypotheses are as:

$$\begin{aligned} \mathcal{H}_0 &: \boldsymbol{x}[n] = \boldsymbol{w}[n] \qquad , n = 0, \cdots, N-1 \\ \mathcal{H}_1 &: \boldsymbol{x}[n] = \mu + \boldsymbol{w}[n], n = 0, \cdots, N-1 \end{aligned} \tag{5}$$

where μ is a known positive constant and $\boldsymbol{w}[n]$ is a Gaussian white noise sequence with mean zero and known variance σ^2.

This section gives the mathematical derivation of Neyman-Pearson (N-P) Likelihood Ratio Test (LRT) to choose between $\mathcal{H}_0$ and $\mathcal{H}_1$ using a fixed N number of samples for a given P_{FA}. The likelihood function is given as,

$$\begin{aligned} L(\boldsymbol{x}[n]) &= \frac{\exp\left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(\boldsymbol{x}[n]-\mu)^2\right]}{\exp\left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2\right]} \\ &= \exp\left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(\boldsymbol{x}[n]-\mu)^2 + \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2\right] \\ &= \exp\left[\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}\{-(\boldsymbol{x}[n]-\mu)^2 + (\boldsymbol{x}[n])^2\}\right] \\ &= \exp\left[\frac{\mu}{2\sigma^2}\sum_{n=0}^{N-1}\{-\mu + 2\boldsymbol{x}[n]\}\right] \end{aligned}$$

Now decide $\mathcal{H}_0$ if,

$$\frac{\mu}{2\sigma^2}\sum_{n=0}^{N-1}\{-\mu+2\boldsymbol{x}[n]\}\geq \ln(\gamma)$$
$$-\mu N+2\sum_{n=0}^{N-1}\boldsymbol{x}[n]\geq \frac{2\sigma^2}{\mu}\ln(\gamma)$$
$$\sum_{n=0}^{N-1}\boldsymbol{x}[n]\geq \frac{\sigma^2}{\mu}\ln(\gamma)+\mu\frac{N}{2}=\gamma^{'} \quad (6)$$

Here $\gamma^{'}$ is the new threshold. In NP method, we choose the desired P_{FA} and accordingly determine the threshold $\gamma^{'}$. Now P_D and P_{FA} is defined as,

$$P_{FA}=\int_{\gamma^{'}}^{\infty}p(\boldsymbol{x}[n];\mathcal{H}_0)d\boldsymbol{x}[n]$$
$$P_D=\int_{\gamma^{'}}^{\infty}p(\boldsymbol{x}[n];\mathcal{H}_1)d\boldsymbol{x}[n],$$

also,

$$E(T(\boldsymbol{x}[n]);\mathcal{H}_0)=E\left(\sum_{n=0}^{N-1}\boldsymbol{w}[n]\right)=0$$
$$E(T(\boldsymbol{x}[n]);\mathcal{H}_1)=E\left(\sum_{n=0}^{N-1}(\mu+\boldsymbol{w}[n])\right)=N\mu,$$

and

$$var(T(\boldsymbol{x}[n]);\mathcal{H}_0)=var\left(\sum_{n=0}^{N-1}\boldsymbol{w}[n]\right)=N\sigma^2$$
$$var(T(\boldsymbol{x}[n]);\mathcal{H}_1)=var\left(\sum_{n=0}^{N-1}(\mu+\boldsymbol{w}[n])\right)=N\sigma^2.$$

Therefore solving,

$$P_{FA}=\Pr\left\{T(\boldsymbol{x}[n])>\gamma^{'};\mathcal{H}_0\right\},$$
$$P_{FA}=Q\left(\frac{\gamma^{'}}{\sqrt{N\sigma^2}}\right),$$
$$Q^{-1}(P_{FA})=\frac{\gamma^{'}}{\sqrt{N\sigma^2}},$$
$$\gamma^{'}=\sqrt{N\sigma^2}Q^{-1}(P_{FA})$$

where $Q(x)$ is referred to as the complementary cumulative distribution function and as defined as,

$$Q(x)=\int_x^{\infty}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}t^2\right)dt.$$

and,

$$P_D=\Pr\left\{T(\boldsymbol{x}[n])>\gamma^{'};\mathcal{H}_1\right\},$$
$$P_D=Q\left(\frac{\gamma^{'}-N\mu}{\sqrt{N\sigma^2}}\right),$$
$$P_D=Q\left(\frac{\sqrt{N\sigma^2}Q^{-1}(P_{FA})-N\mu}{\sqrt{N\sigma^2}}\right),$$
$$P_D=Q\left(Q^{-1}(P_{FA})-\sqrt{\frac{N\mu^2}{\sigma^2}}\right).$$

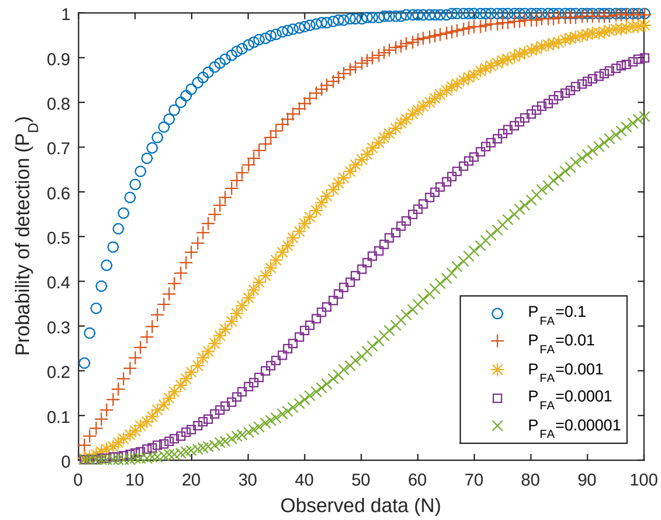

Fig. 2. N-P test: plot of P_D against N for case I

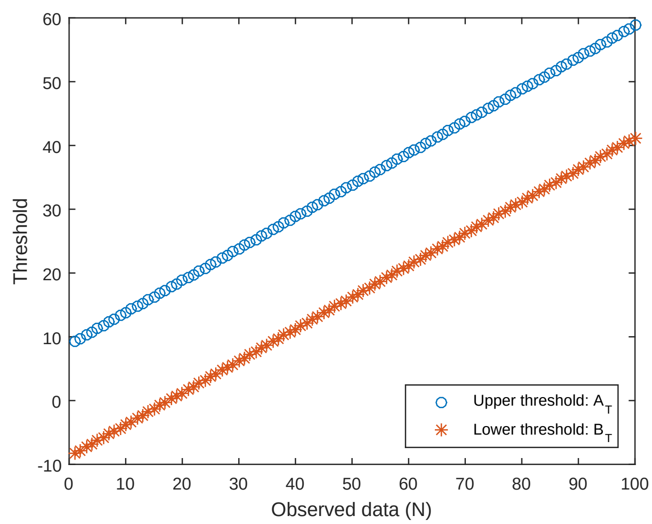

Fig. 3. Sequential detection: plot of thresholds against observed data for case I

In particular, for $\mu=1$ and $\sigma^2=4$ the plot of P_D as a function of N is given in Fig. 2 for various values of P_{FA}.

Now in sequential detection we deal with two thresholds - upper and lower. If the data samples lead to the upper threshold, we choose hypothesis $\mathcal{H}_1$. If the data samples lead to the lower threshold, we choose hypothesis $\mathcal{H}_0$. Otherwise, a data sample is added to reach either threshold as described in section II. This test can easily be written using (6). The upper and lower threshold are respectively A_T and B_T as follows:

$$A_T=\frac{\sigma^2}{\mu}\ln(A)+\mu\frac{N}{2},$$
$$B_T=\frac{\sigma^2}{\mu}\ln(B)+\mu\frac{N}{2}.$$

where A and B are given as in (3) and (4).

Fig. 3 shows the plot of both thresholds against N for $P_{FA} = 0.1$, $P_D = 0.9$, $\mu = 1$ and $\sigma^2 = 4$. The plot shows that the two thresholds are linear functions of N. This is because of the way we have chosen our test statistic. So as the data points increases, the threshold increases accordingly. The following derivation will lead to a threshold test that can be computed recursively. From (6):

$$\sum_{n=0}^{N-1} \boldsymbol{x}[n] \geq \frac{\sigma^2}{\mu} \ln(\gamma) + \mu \frac{N}{2},$$

$$\boldsymbol{x}[N] \geq \frac{\sigma^2}{\mu} \ln(\gamma) + \mu \frac{N}{2} - \sum_{n=0}^{N-1} \boldsymbol{x}[n] + \boldsymbol{x}[N]$$

2) Case II: The two hypotheses are described as:

$$\mathcal{H}_0 : \boldsymbol{x}[n] = \boldsymbol{w}[n] \qquad , n = 0, \cdots, N-1$$
$$\mathcal{H}_1 : \boldsymbol{x}[n] = \boldsymbol{s}[n] + \boldsymbol{w}[n], n = 0, \cdots, N-1$$

where $\boldsymbol{w}[n]$ is a Gaussian white noise sequence with mean zero and known variance σ^2 and $\boldsymbol{s}[n]$ is a sinusoidal signal:

$$\boldsymbol{s}[n] = \mu \cos(\omega n), \qquad -\pi \leq \omega \leq \pi$$

This section gives the mathematical derivation of N-P LRT to choose between $\mathcal{H}_0$ and $\mathcal{H}_1$ using a fixed number of samples N for a given P_{FA}. As always the likelihood ratio is given as,

$$L(\boldsymbol{x}[n]) = \frac{\exp\left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(\boldsymbol{x}[n] - \mu\cos(\omega n))^2\right]}{\exp\left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2\right]}$$
$$= \exp\left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(\boldsymbol{x}[n] - \mu\cos(\omega n))^2 + \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2\right]$$
$$= \exp\left[\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}\left\{-(\boldsymbol{x}[n] - \mu\cos(\omega n))^2 + (\boldsymbol{x}[n])^2\right\}\right]$$
$$= \exp\left[\frac{\mu}{2\sigma^2}\sum_{n=0}^{N-1}\{-\mu\cos(\omega n) + 2\boldsymbol{x}[n]\}\cos(\omega n)\right]$$

Now we compare $L(x[n])$ to threshold (γ) to decide $\mathcal{H}_0$, and take the natural logarithm on both sides which yields,

$$\frac{\mu}{2\sigma^2}\sum_{n=0}^{N-1}\{-\mu\cos(\omega n) + 2\boldsymbol{x}[n]\}\cos(\omega n) \geq \ln(\gamma)$$

$$\sum_{n=0}^{N-1}\boldsymbol{x}[n]\cos(\omega n) \geq \gamma' \tag{7}$$

where $\gamma' = \frac{\sigma^2}{\mu}\ln(\gamma) + \frac{\mu}{2}\sum_{n=0}^{N-1}\cos^2(\omega n)$ is the new threshold. In N-P method, we choose the desired P_{FA} and accordingly determine the threshold.

In this case also, for sequential detection we deal with two thresholds upper and lower. If the data samples lead to the upper threshold, we choose hypothesis $\mathcal{H}_1$. If the data samples lead to the lower threshold, we choose hypothesis $\mathcal{H}_0$. Otherwise, a data sample is added to reach either threshold. This test can easily be written using (7). The upper and lower threshold are respectively A_T and B_T as follows:

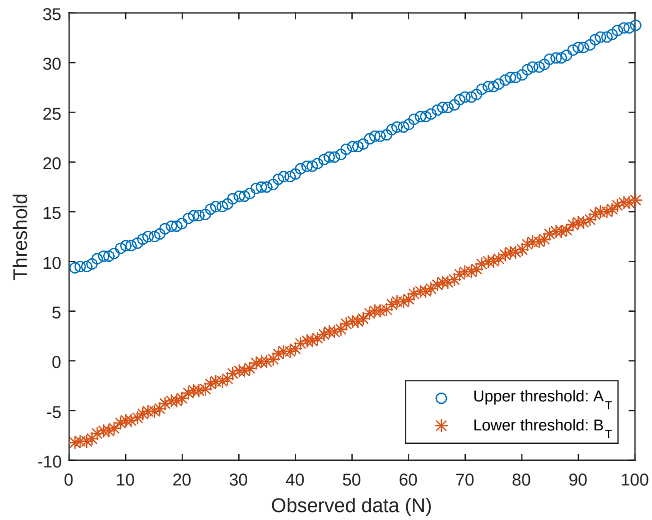

Fig. 4. Sequential detection: plot of thresholds against observed data for case II

$$A_T = \frac{\sigma^2}{\mu}\ln(A) + \frac{\mu}{2}\sum_{n=0}^{N-1}\cos^2(\omega n),$$

$$B_T = \frac{\sigma^2}{\mu}\ln(B) + \frac{\mu}{2}\sum_{n=0}^{N-1}\cos^2(\omega n).$$

Representing recursively we have:

$$\boldsymbol{x}[N] \geq \frac{\sigma^2}{\mu}\ln(\gamma) + \frac{\mu}{2}\sum_{n=0}^{N-1}\cos^2(\omega n) - \sum_{n=0}^{N-1}\boldsymbol{x}[n]\cos(\omega n) + \boldsymbol{x}[N]$$

Fig. 4 shows the plot of both thresholds against N for $P_{FA} = 0.1$, $P_D = 0.9$, $\mu = 1$, $\sigma^2 = 4$ and $\omega = \frac{2\pi}{8}$. The plot shows that the two thresholds linear changes with N.

3) Case III: The two hypotheses are described as:

$$\mathcal{H}_0 : \boldsymbol{x}[n] = \boldsymbol{w}[n] \qquad , n = 0, \cdots, N-1$$
$$\mathcal{H}_1 : \boldsymbol{x}[n] = \boldsymbol{s}[n] + \boldsymbol{w}[n], n = 0, \cdots, N-1$$

where $\boldsymbol{w}[n]$ is Gaussian white noise with zero mean and known variance σ_N^2 and the signal $\boldsymbol{s}[n]$ is a white Gaussian sequence with zero mean and known variance σ_S^2. $\boldsymbol{s}[n]$ and $\boldsymbol{w}[n]$ are uncorrelated to each other.

This section is the mathematical derivation of N-P LRT to choose between $\mathcal{H}_0$ and $\mathcal{H}_1$ using a fixed number of samples

N for a given P_{FA}. The likelihood function is given as,

$$L(\boldsymbol{x}[n]) = \frac{\left(\frac{1}{\sqrt{2\pi(\sigma_N^2+\sigma_S^2)}}\right)^N e^{\left[-\frac{1}{2(\sigma_N^2+\sigma_S^2)}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2\right]}}{\left(\frac{1}{\sqrt{2\pi\sigma_N^2}}\right)^N e^{\left[-\frac{1}{2\sigma_N^2}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2\right]}}$$
$$= \left(\frac{\sigma_N^2}{\sigma_N^2+\sigma_S^2}\right)^{\frac{N}{2}} \frac{\exp\left[-\frac{1}{2(\sigma_N^2+\sigma_S^2)}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2\right]}{\exp\left[-\frac{1}{2\sigma_N^2}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2\right]}$$
$$= \left(\frac{\sigma_N^2}{\sigma_N^2+\sigma_S^2}\right)^{\frac{N}{2}} \exp\left[-\frac{1}{2(\sigma_N^2+\sigma_S^2)}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2 + \frac{1}{2\sigma_N^2}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2\right]$$
$$= \left(\frac{\sigma_N^2}{\sigma_N^2+\sigma_S^2}\right)^{\frac{N}{2}} \exp\left[\frac{\sigma_S^2}{2\sigma_N^2(\sigma_N^2+\sigma_S^2)}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2\right]$$

Now we compare $L(x[n])$ to decide to threshold γ, and take the natural logarithm on both sides.

$$\frac{N}{2}\ln\left(\frac{\sigma_N^2}{\sigma_N^2+\sigma_S^2}\right) + \left[\frac{\sigma_S^2}{2\sigma_N^2(\sigma_N^2+\sigma_S^2)}\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2\right] \geq \ln(\gamma)$$

$$\sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2 \geq \frac{2\sigma_N^2(\sigma_N^2+\sigma_S^2)}{\sigma_S^2}\left[\ln(\gamma) + \frac{N}{2}\ln\left(\frac{\sigma_N^2+\sigma_S^2}{\sigma_N^2}\right)\right] = \gamma^{'} \tag{8}$$

where, $\gamma^{'}$ is the new threshold. In N-P method, we choose the desired P_{FA} and accordingly determine the threshold. P_D and P_{FA} is defined as,

$$P_{FA} = \int_{\gamma^{'}}^{\infty} p(\boldsymbol{x}[n];\mathcal{H}_0)d\boldsymbol{x}[n]$$
$$P_D = \int_{\gamma^{'}}^{\infty} p(\boldsymbol{x}[n];\mathcal{H}_1)d\boldsymbol{x}[n],$$

also, for $\sigma_S^2 = 1$ and $\sigma_N^2 = 1$, we get

$$E(T(\boldsymbol{x}[n]);\mathcal{H}_0) = E\left(\sum_{n=0}^{N-1}(\boldsymbol{w}[n])^2\right) = \sum_{n=0}^{N-1}E(\boldsymbol{w}[n])^2$$
$$= \sum_{n=0}^{N-1}(var(\boldsymbol{w}[n]) + E(\boldsymbol{w}[n])^2) = N$$
$$E(T(\boldsymbol{x}[n]);\mathcal{H}_1) = E\left(\sum_{n=0}^{N-1}(\boldsymbol{s}[n]+\boldsymbol{w}[n])^2\right) = 2N$$

and

$$var(T(\boldsymbol{x}[n]);\mathcal{H}_0) = var\left(\sum_{n=0}^{N-1}(\boldsymbol{w}[n])^2\right) = 2N$$
$$var(T(\boldsymbol{x}[n]);\mathcal{H}_1) = var\left(\sum_{n=0}^{N-1}(\boldsymbol{s}[n]+\boldsymbol{w}[n])^2\right) = 4N.$$

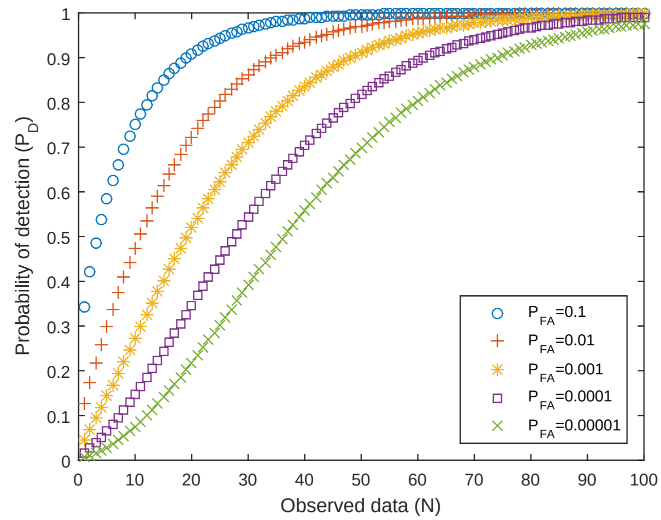

Fig. 5. N-P test: plot of P_D against N for case III

Therefore solving,

$$P_{FA} = \Pr\left\{T(\boldsymbol{x}[n]) > \gamma^{'};\mathcal{H}_0\right\},$$
$$P_{FA} = Q\left(\frac{\gamma^{'}-N}{\sqrt{2N}}\right),$$
$$Q^{-1}(P_{FA}) = \frac{\gamma^{'}-N}{\sqrt{2N}},$$
$$\gamma^{'} = \sqrt{2N}Q^{-1}(P_{FA}) + N,$$

and,

$$P_D = \Pr\left\{T(\boldsymbol{x}[n]) > \gamma^{'};\mathcal{H}_1\right\},$$
$$P_D = Q\left(\frac{\gamma^{'}-2N}{\sqrt{4N}}\right),$$
$$P_D = Q\left(\frac{\sqrt{2N}Q^{-1}(P_{FA}) + N - 2N}{\sqrt{4N}}\right),$$
$$P_D = Q\left(\frac{Q^{-1}(P_{FA})}{\sqrt{2}} - \frac{\sqrt{N}}{2}\right).$$

For $\sigma_S^2 = 1$, $\sigma_N^2 = 1$ the plot of PD as a function of N is given in Fig. 5 for several values of P_{FA}.

In sequential detection we deal with two thresholds as always - upper and lower. If the data samples lead to the upper threshold, we choose hypothesis $\mathcal{H}_1$. If the data samples lead to the lower threshold, we choose hypothesis $\mathcal{H}_0$. Otherwise, a data sample is added to reach either threshold. This test can easily be written using (8). The upper and lower threshold are respectively A_T and B_T as follows:

$$A_T = \frac{2\sigma_N^2(\sigma_N^2+\sigma_S^2)}{\sigma_S^2}\left[\ln(A) + \frac{N}{2}\ln\left(\frac{\sigma_N^2+\sigma_S^2}{\sigma_N^2}\right)\right],$$
$$B_T = \frac{2\sigma_N^2(\sigma_N^2+\sigma_S^2)}{\sigma_S^2}\left[\ln(B) + \frac{N}{2}\ln\left(\frac{\sigma_N^2+\sigma_S^2}{\sigma_N^2}\right)\right].$$

Representing recursively gives:

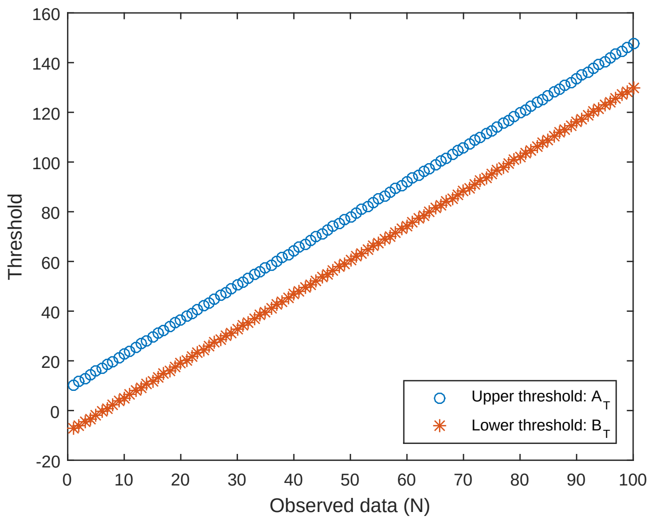

Fig. 6. Sequential detection: plot of thresholds against observed data for case III

$$\boldsymbol{x}[N] \geq \frac{2\sigma_N^2(\sigma_N^2+\sigma_S^2)}{\sigma_S^2}\left[\ln(A) + \frac{N}{2}\ln\left(\frac{\sigma_N^2+\sigma_S^2}{\sigma_N^2}\right)\right] - \sum_{n=0}^{N-1}(\boldsymbol{x}[n])^2 + \boldsymbol{x}[N]$$

Fig. 6 shows the typical plot of both thresholds against N for $P_{FA} = 0.1$, $P_D = 0.9$, $\mu = 1$, $\sigma_N^2 = 1$ and $\sigma_S^2 = 1$. The plot shows that the two thresholds linear changes with N.

V. Test Results

In this section we will discuss the test results received from computer simulation for all the cases. MATLAB methods used for the simulation are included as a separate section.

As discussed in section II, the test statistic is compared to upper and lower thresholds starting from data sample $x[0]$. If the test statistic exceeds the upper threshold, then $\mathcal{H}_1$ is decided. If the test statistic falls below the lower threshold, then $\mathcal{H}_0$ is decided. If the test statistic remains in between the two thresholds, then another data sample is added to the test statistic and the procedure is repeated. With each data sample added to the test, the test statistic and the thresholds are altered. This is done several hundred times (iterations). For P_{FA} the data samples are generated using the density under hypothesis $\mathcal{H}_0$. At the end of all the iterations, the number of times our signal exceeds upper threshold is divided by total number of iterations to get the desired P_{FA}. For P_D, the data samples are generated under hypothesis $\mathcal{H}_1$. At the end of all iterations, the number of times our signal exceeds upper threshold is divided by total number of iterations to get the desired P_D.

For case I, according to the simulations, 18 samples are needed on average to get a P_{FA} of 0.1004 and 16 samples are needed on average to get a P_D of 0.9240. Obtaining the same P_{FA} and P_D with fixed length decision rule would require about 25 samples (as seen from Fig. 2). So the number of data samples is reduced using sequential detection.

Likewise in case II, on average 32 samples are needed to get a P_{FA} of 0.085 and 32 samples are needed to get a P_D of 0.909. We see that a larger set of data was needed to get values that were even less accurate as compared to the previous case of part I. This is because in the previous case we had a DC signal instead of a sinusoidal. The magnitude of the sinusoidal takes value between negative and positive one, thus changing the mean of our signal with each new data sample. As this changing mean in-fact also pass through the mean of noise, it is more difficult to distinguish between the two. Therefore, even to get a less accurate answer we need more data samples.

In case III on average 23 samples are needed to get a P_{FA} of 0.0815. On the other hand 15 samples are needed to get a P_D of 0.9368. From Fig. 5 it is seen that about 30 data samples are needed in the case of fixed length decision rule. It is thus clear that a smaller data set is needed in the case of sequential detection.

VI. Conclusion

It is clear from the test result, the number of data samples is reduced using sequential detection. This reduced number of data samples is what we expected with the sequential detection because the process goes as follows: firstly, sequential detection tries to get the decision rule from a single data sample. If a decision is not reached, another sample (feature) is included in the test, and so on. Eventually, the smallest possible subset of data samples that converges to a decision is attained and hence the decision is made. The important thing is that we rely on this smallest possible 'decision making' subset of data. Whereas, in the case of fixed length decision rule, we usually have more data than required to reach a decision.

References

[1] S. M. Kay, *Fundamentals of statistical signal processing: Detection theory, vol. 2*. Prentice Hall Upper Saddle River, NJ, USA:, 1998.

[2] V. Tuzlukov, *Signal Processing in Radar Systems*. Taylor & Francis, 2012. [Online]. Available: https://books.google.com/books?id=EWjO_kAUEhoC

[3] I. Selin, "The sequential estimation and detection of signals in normal noise. i," *Information and Control*, vol. 7, no. 4, pp. 512 – 534, 1964.

[4] P. Sheu, H. Yu, C. Ramamoorthy, A. Joshi, and L. Zadeh, *Semantic Computing*. Wiley, 2011. [Online]. Available: https://books.google.com/books?id=C_NiygT_RcoC

MATLAB codes for: A Case Study of Basic Sequential Detection Methods

Arindam Bose

April 26, 2016

1 MATLAB Code: case_I_p1.m

```
% Description: Program to plot Probability of detection (P_D) as a
%              function of data points (N) for different Probability
%              of false alarm (P_FA) for case I
% Author:      Arindam Bose
% Date:        04/26/2016
% Course:      Estimation and Detection theory (ECE 531)

N = 100;
n = 1:1:N;

mn = 1;
variance = 4;
PFA = [0.1 0.01 0.001 0.0001 0.00001];
pointer = ['o' '+' '*' 's' 'x'];

for i = 1:size(PFA,2)
    PD = Q(Qinv(PFA(i)) - sqrt(n * mn^2 / variance));
    plot(n,PD,pointer(i));
    hold on;
end

xlabel('Observed data (N)');
ylabel('Probability of detection (P_D)');
legend('P_{FA}=0.1','P_{FA}=0.01','P_{FA}=0.001','P_{FA}=0.0001',...
    'P_{FA}=0.00001','Location','southeast');
```

2 MATLAB Code: case_I_p2.m

```
% Description: Program to plot the change of Upper threshold (A_T) and
%              Lower threshold (B_T) with change in data points (N)
%              for sequential detection method for case I
% Author:      Arindam Bose
% Date:        04/26/2016
% Course:      Estimation and Detection theory (ECE 531)

N = 100;
```

```
n = 1:1:N;

PD = 0.9;
PFA = 0.1;
mn = 1;
variance = 4;

A = PD/PFA;
B = (1-PD)/(1-PFA);

A_T = variance * log(A) / mn + mn*n/2;
B_T = variance * log(B) / mn + mn*n/2;

plot(n,A_T,'o')
hold on
plot(n,B_T,'*')

xlabel('Observed data (N)');
ylabel('Threshold');
legend('Upper threshold: A_T','Lower threshold: B_T',...
    'Location','southeast');
```

3 MATLAB Code: case_II_p2.m

```
% Description: Program to plot the change of Upper threshold (A_T) and
%              Lower threshold (B_T) with change in data points (N)
%              for sequential detection method for case II
% Author:      Arindam Bose
% Date:        04/26/2016
% Course:      Estimation and Detection theory (ECE 531)

N = 100;
n = 1:1:N;

PD = 0.9;
PFA = 0.1;
mn = 1;
variance = 4;

A = PD/PFA;
B = (1-PD)/(1-PFA);

for i = 1:1:N
    A_T(i) = variance*log(A)/mn + mn/2*sum(cos(2*pi*(0:i-1)/8).^2);
    B_T(i) = variance*log(B)/mn + mn/2*sum(cos(2*pi*(0:i-1)/8).^2);
end

plot(n,A_T,'o')
hold on
plot(n,B_T,'*')

```

```
xlabel('Observed data (N)');
ylabel('Threshold');
legend('Upper threshold: A_T','Lower threshold: B_T',...
    'Location','southeast');
```

4 MATLAB Code: case_III_p1.m

```
% Description: Program to plot Probability of detection (P_D) as a
%              function of data points (N) for different Probability
%              of false alarm (P_FA) for case III
% Author:      Arindam Bose
% Date:        04/26/2016
% Course:      Estimation and Detection theory (ECE 531)

N = 100;
n = 1:1:N;

PFA = [0.1 0.01 0.001 0.0001 0.00001];
pointer = ['o' '+' '*' 's' 'x'];

for i = 1:size(PFA,2)
    PD = Q(Qinv(PFA(i))/sqrt(2) - sqrt(n)/2);
    plot(n,PD,pointer(i));
    hold on;
end

xlabel('Observed data (N)');
ylabel('Probability of detection (P_D)');
legend('P_{FA}=0.1','P_{FA}=0.01','P_{FA}=0.001',...
    'P_{FA}=0.0001','P_{FA}=0.00001','Location','southeast');
```

5 MATLAB Code: case_III_p2.m

```
% Description: Program to plot the change of Upper threshold (A_T) and
%              Lower threshold (B_T) with change in data points (N)
%              for sequential detection method for case III
% Author:      Arindam Bose
% Date:        04/26/2016
% Course:      Estimation and Detection theory (ECE 531)

N = 100;
n = 1:1:N;

PD = 0.9;
PFA = 0.1;
variance = 4;

A = PD/PFA;
```

```
B = (1-PD)/(1-PFA);

A_T = 4*(log(A) + n/2*log(2));
B_T = 4*(log(B) + n/2*log(2));

plot(n,A_T,'o')
hold on
plot(n,B_T,'*')

xlabel('Observed data (N)');
ylabel('Threshold');
legend('Upper threshold: A_T','Lower threshold: B_T',...
    'Location','southeast');
```

6 MATLAB Code: sd_pd_case_I.m

```
% Description: Method to estimate Probability of detection (P_D)
%              and the average number of data points (for each
%              iteration) needed to reach the pd in case I.
% Author:      Arindam Bose
% Date:        04/26/2016
% Course:      Estimation and Detection theory (ECE 531)

clear; clc;
iterations = 1000; % Number of trials

A_const = 1;
WGN_mean = 0; % Mean of the white noise
WGN_var = 4; % Variance of the white noise
mean_H1 = A_const + WGN_mean; % Mean of data, given H1
var_H1 = WGN_var; % Variance of data, given H1

estimated_pd = 0;
N_avg = 0;
N_total = 0;
D = 0;  % Number of times Detection occurs

for m = 1:iterations  % m denotes the total number of iterations
    N_trial = 0; % Counter for recording number of times while loop executes
    A_thresh = 4*log(9) + N_trial/2;
    B_thresh = 4*log(1/9) + N_trial/2;
    sum_X = mean_H1 + randn(1,1)*sqrt(var_H1);

    while ((sum_X > B_thresh) & (sum_X < A_thresh))
        N_trial = N_trial + 1;
        sum_X = sum_X + mean_H1 + randn(1,1)*sqrt(var_H1);
        A_thresh = 4*log(9) + N_trial/2;
        B_thresh = 4*log(1/9) + N_trial/2;
    end

```

```
    N_total = N_total + (N_trial + 1);

    if sum_X >= A_thresh
        D = D + 1;
    end
end

N_avg = N_total/m
estimated_pd = D/m
```

7 MATLAB Code: sd_pd_case_II.m

```
% Description: Method to estimate Probability of detection (P_D)
%              and the average number of data points (for each
%              iteration) needed to reach the pd in case II.
% Author:      Arindam Bose
% Date:        04/26/2016
% Course:      Estimation and Detection theory (ECE 531)

clear; clc;
iterations = 1000; % Number of trials

A_const = 1;
WGN_mean = 0; % Mean of the white noise
WGN_var = 4; % Variance of the white noise
mean_H1 = A_const + WGN_mean; % Mean of data, given H1
var_H1 = WGN_var; % Variance of data, given H1

estimated_pd = 0;
N_avg = 0;
N_total = 0;
D = 0; % Number of times Detection occurs

for m = 1:iterations % m denotes the total number of iterations
    N_trial = 0; % Counter for recording number of times while loop
        executes
    A_thresh = 4 * log(9) + (cos(2 * pi * N_trial / 8))^2 / 2;
    B_thresh = 4 * log(1/9) + (cos(2 * pi * N_trial / 8))^2 / 2;
    sum_X = (cos(2*pi*N_trial/8) + randn(1,1) * sqrt(var_H1)) * cos(2*pi*
        N_trial/8);
%   sum_X = mean_H1 + randn(1,1) * sqrt(var_H1);


    while ((sum_X > B_thresh) & (sum_X < A_thresh))
        N_trial = N_trial + 1;
        sum_X = sum_X + (cos(2*pi*N_trial/8) + randn(1,1) * sqrt(var_H1))
            * cos(2 * pi * N_trial / 8);
        A_thresh = A_thresh + (cos(2 * pi * N_trial / 8))^2 / 2;
        B_thresh = B_thresh + (cos(2 * pi * N_trial / 8))^2 / 2;
    end

```

```
    N_total = N_total + (N_trial + 1);

    if sum_X >= A_thresh
        D = D + 1;
    end
end

N_avg = N_total / m
estimated_pd = D/m
```

8 MATLAB Code: sd_pd_case_III.m

```
% Description: Method to estimate Probability of detection (P_D)
%              and the average number of data points (for each
%              iteration) needed to reach the pd in case III.
% Author:      Arindam Bose
% Date:        04/26/2016
% Course:      Estimation and Detection theory (ECE 531)

clear; clc;
iterations = 1000; % Number of trials

WGN_mean = 0; % Mean of the white noise
WGN_var = 1; % Variance of the white noise
mean_H1 = 0 + WGN_mean; % Mean of data, given H1
var_H1 = 1 + WGN_var; % Variance of data, given H1

estimated_pd = 0;
N_avg = 0;
N_total = 0;
D = 0;  % Number of times Detection occurs

for m = 1:iterations % m denotes the total number of iterations
    N_trial = 0; % Counter for recording number of times while loop
       executes
    A_thresh = 4 * log(9) + 2*N_trial*log(2);
    B_thresh = 4 * log(1/9) + 2*N_trial*log(2);
    sum_X = (mean_H1 + randn(1,1) * sqrt(var_H1))^2;

    while ((sum_X > B_thresh) & (sum_X < A_thresh))
        N_trial = N_trial + 1;
        sum_X = sum_X + (mean_H1 + randn(1,1) * sqrt(var_H1))^2;
        A_thresh = 4 * log(9) + 2*N_trial*log(2);
        B_thresh = 4 * log(1/9) + 2*N_trial*log(2);
    end

    N_total = N_total + (N_trial + 1);

    if sum_X >= A_thresh
        D = D + 1;
    end
```

```
end

N_avg = N_total / m
estimated_pd = D/m
```

9 MATLAB Code: sd_pfa_case_I.m

```
% Description: Method to estimate Probability of false alarm (P_FA)
%              and the average number of data points (for each
%              iteration) needed to reach the pfa in case I.
% Author:      Arindam Bose
% Date:        04/26/2016
% Course:      Estimation and Detection theory (ECE 531)
clear; clc;
iterations = 1000; % Number of trials

WGN_mean = 0; % Mean of the white noise
WGN_var = 4; % Variance of the white noise
mean_H0 = WGN_mean; % Mean of data, given H0
var_H0 = WGN_var; % Variance of data, given H0

estimated_pfa = 0;
N_avg = 0;
N_total = 0;
FA = 0; % Number of times False Alarm occurs

for m = 1:iterations % m denotes the total number of iterations
    N_trial = 0; % Counter for recording number of times while loop
       executes
    A_thresh = 4*log(9) + N_trial/2;
    B_thresh = 4*log(1/9) + N_trial/2;
    sum_X = mean_H0 + randn(1,1)*sqrt(var_H0);

    while ((sum_X > B_thresh) & (sum_X < A_thresh))
        N_trial = N_trial + 1;
        sum_X = sum_X + mean_H0 + randn(1,1)*sqrt(var_H0);
        A_thresh = 4*log(9) + N_trial/2;
        B_thresh = 4*log(1/9) + N_trial/2;
    end

    N_total = N_total + (N_trial + 1);

    if sum_X >= A_thresh
        FA = FA + 1;
    end
end

N_avg = N_total/m
estimated_pfa = FA/m
```

10 MATLAB Code: sd_pfa_case_II.m

```
% Description: Method to estimate Probability of false alarm (P_FA)
%              and the average number of data points (for each
%              iteration) needed to reach the pfa in case II.
% Author:      Arindam Bose
% Date:        04/26/2016
% Course:      Estimation and Detection theory (ECE 531)

clear; clc;
iterations = 1000; % Number of trials

WGN_mean = 0; % Mean of the white noise
WGN_var = 4; % Variance of the white noise
mean_H0 = WGN_mean; % Mean of data, given H0
var_H0 = WGN_var; % Variance of data, given H0

estimated_pfa = 0;
N_avg = 0;
N_total = 0;
FA = 0; % Number of times False Alarm occurs

for m = 1:iterations % m denotes the total number of iterations
    N_trial = 0; % Counter for recording number of times while loop
        executes
    A_thresh = 4 * log(9) + (cos(2 * pi * N_trial / 8))^2 / 2;
    B_thresh = 4 * log(1/9) + (cos(2 * pi * N_trial / 8))^2 / 2;
    sum_X = (mean_H0 + randn(1,1) * sqrt(var_H0)) * cos(2 * pi * N_trial /
        8);

    while ((sum_X > B_thresh) & (sum_X < A_thresh))
        N_trial = N_trial + 1;
        sum_X = sum_X + (mean_H0 + randn(1,1) * sqrt(var_H0)) * cos(2 * pi
            * N_trial / 8);
        A_thresh = A_thresh + (cos(2 * pi * N_trial / 8))^2 / 2;
        B_thresh = B_thresh + (cos(2 * pi * N_trial / 8))^2 / 2;
    end

    N_total = N_total + (N_trial + 1);

    if sum_X >= A_thresh
        FA = FA + 1;
    end
end

N_avg = N_total / m
estimated_pfa = FA / m
```

11 MATLAB Code: sd_pfa_case_III.m

```
% Description: Method to estimate Probability of false alarm (P_FA)
```

```
%               and the average number of data points (for each
%               iteration) needed to reach the pfa in case III.
% Author:       Arindam Bose
% Date:         04/26/2016
% Course:       Estimation and Detection theory (ECE 531)

clear; clc;
iterations = 10000; % Number of trials

WGN_mean = 0; % Mean of the white noise
WGN_var = 1; % Variance of the white noise
mean_H0 = WGN_mean; % Mean of data, given H0
var_H0 = WGN_var; % Variance of data, given H0

estimated_pfa = 0;
N_avg = 0;
N_total = 0;
FA = 0; % Number of times False Alarm occurs

for m = 1:iterations % m denotes the total number of iterations
    N_trial = 0; % Counter for recording number of times while loop executes
    A_thresh = 4 * log(9) + 2*N_trial*log(2);
    B_thresh = 4 * log(1/9) + 2*N_trial*log(2);
    sum_X = (mean_H0 + randn(1,1) * sqrt(var_H0))^2;

    while ((sum_X > B_thresh) & (sum_X < A_thresh))
        N_trial = N_trial + 1;
        A_thresh = 4 * log(9) + 2*N_trial*log(2);
        B_thresh = 4 * log(1/9) + 2*N_trial*log(2);
        sum_X = sum_X + (mean_H0 + randn(1,1) * sqrt(var_H0))^2;
    end

    N_total = N_total + (N_trial + 1);

    if sum_X >= A_thresh
        FA = FA + 1;
    end
end

N_avg = N_total / m
estimated_pfa = FA / m
```

12 MATLAB Code: Q.m

```
  function y=Q(x)
% This program computes the right-tail probability
% (complementary cumulative distribution function) for
% a N(0,1) random variable.
%
% Input Parameters:
```

```
%
%  x - Rreal column vector of x values
%
% Output Parameters:
%
%  y - Real column vector of right-tail probabilities
%
% Verification Test Case:
%
% The input x=[0 1 2]'; should produce y=[0.5 0.1587 0.0228]'.
%
  y=0.5*erfc(x/sqrt(2));
```

13 MATLAB Code: Qinv.m

```
  function y=Qinv(x)
% This program computes the inverse Q function or the value
% which is exceeded by a N(0,1) random variable with a
% probability of x.
%
% Input Parameters:
%
%  x - Real column vector of right-tail probabilities
%      (in intercal [0,1])
%
% Output Parameters:
%
%  y - Real column vector of values of random variable
%
% Verification Test Case:
%
% The input x=[0.5 0.1587 0.0228]'; should produce
% y=[0 0.9998 1.9991]'.
%
  y=sqrt(2)*erfinv(1-2*x);
```