EEE 556 PROJECT
DUE: FEBRUARY 20, 2002

SEQUENTIAL DETECTION

INSTRUCTOR: DR. DARRYL MORRELL

SUBMITTED BY:

IMTIAZ NIZAMI - 993 21 6600

CONTENTS

INTRODUCTION

This project familiarizes us with the theory of sequential detection. The performance of a detector can be improved by increasing the signal-to-noise ratio. In most cases, the noise power is fixed, and in order to increase the signal-to-noise ratio we have to increase the signal energy. 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. 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 this project, sequential detection scheme is implemented for three different cases. In each case we deal with two hypotheses. Hypothesis H_0 (null hypothesis) is WGN (white Gaussian noise) in all cases. A brief description of these three cases is as follows.

I. Hypothesis H_0 is WGN with mean of zero and variance of four. The hypothesis H_1 is the sum of WGN and a DC signal 'A' of magnitude of one.

II. Hypothesis H_0 is WGN with mean of zero and variance of four. The hypothesis H_1 is the sum of WGN and a cosine function of the form "$A\cos(2\pi n/8)$". Here 'A' is a constant of magnitude one.

III. Hypothesis H_0 is WGN with mean of zero and variance of one. The hypothesis H_1 is the sum of WGN and a white Gaussian sequence with mean equal to zero, and variance of one. H_0 and H_1 are independent of each other.

PART I

The two hypotheses given to us are

$H_0 : x[n] = w[n], \qquad n = 0,\ldots,N-1$

$H_1 : x[n] = A + w[n], \qquad n = 0,\ldots,N-1$

Here 'A' is a positive constant with a value of one. 'w[n]' is WGN with zero mean (μ) and variance (σ^2) of 4.

NEYMAN PEARSON (N-P) LIKELIHOOD RATIO TEST (LRT)

This section gives the mathematical derivation of N-P LRT to choose between H_0 and H_1 using a fixed number of samples 'N' for a given P_{FA}.

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

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

$$L(x[n]) = \exp\left[\frac{1}{8}\sum_{n=0}^{N-1}\left\{-\left(x[n]-1\right)^2 + \left(x[n]\right)^2\right\}\right]$$

$$L(x[n]) = \exp\left[\frac{1}{8}\sum_{n=0}^{N-1}\left\{-\left(x[n]\right)^2 - 1 + 2x[n] + \left(x[n]\right)^2\right\}\right]$$

$$L(x[n]) = \exp\left[\frac{1}{8}\sum_{n=0}^{N-1}\left\{-1 + 2x[n]\right\}\right]$$

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

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

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

$$-N + 2\sum_{n=0}^{N-1}x[n] \geq 8\ln(\gamma)$$

$$2\sum_{n=0}^{N-1} x[n] \geq 8\ln(\gamma) + N$$

$$\sum_{n=0}^{N-1} x[n] \geq 4\ln(\gamma) + N/2 ....(A)$$

Thus

$$\gamma' = 4\ln(\gamma) + N/2$$

Here γ' is the new threshold. In N-P method, we choose the desired P_{FA} (probability of false-alarm) and accordingly determine the threshold (γ').

Now

$$P_{FA} = \int_{\gamma'}^{\infty} p(x[n]; H_0) dx[n]$$

and

$$P_D = \int_{\gamma'}^{\infty} p(x[n]; H_1) dx[n]$$

Also

$$E(T(x[n]; H_0) = E\left(\sum_{n=0}^{N-1} w[n]\right)$$

$$E(T(x[n]; H_0) = 0$$

and

$$E(T(x[n]; H_1) = E\left(\sum_{n=0}^{N-1} (\mu + w[n])\right)$$

$$E(T(x[n]; H_1) = N\mu$$

Also

$$\operatorname{var}(T(x[n]; H_0) = \operatorname{var}\left(\sum_{n=0}^{N-1} w[n]\right)$$

$$\operatorname{var}(T(x[n]; H_0) = N\sigma^2$$

and

$$\operatorname{var}(T(x[n]; H_1) = \operatorname{var}\left(\sum_{n=0}^{N-1} (\mu + w[n])\right)$$

$$\operatorname{var}(T(x[n]; H_1) = N\sigma^2$$

Therefore

$$P_{FA} = \Pr\{T(x[n]) > \gamma'; H_0\}$$

$$P_{FA} = Q\left(\frac{\gamma' - 0}{\sqrt{N\sigma^2}}\right)$$

$$Q^{-1}(P_{FA}) = \frac{\gamma'}{\sqrt{N\sigma^2}}$$

$$\gamma' = \sqrt{N\sigma^2}\,Q^{-1}(P_{FA})$$

and

$$P_D = \Pr\{T(x[n]) > \gamma'; H_1\}$$

$$P_D = Q\left(\frac{\gamma' - N\mu}{\sqrt{N\sigma^2}}\right)$$

$$P_D = Q\left(\frac{\sqrt{N\sigma^2}\,Q^{-1}(P_{FA}) - N\mu}{\sqrt{N\sigma^2}}\right)$$

$$P_D = Q\left(Q^{-1}(P_{FA}) - \sqrt{\frac{N\mu^2}{\sigma^2}}\right)$$

The plot of P_D as a function of 'N' (data points) is given in Figure 1. This plot was generated from the matlab file "partI_ques01".

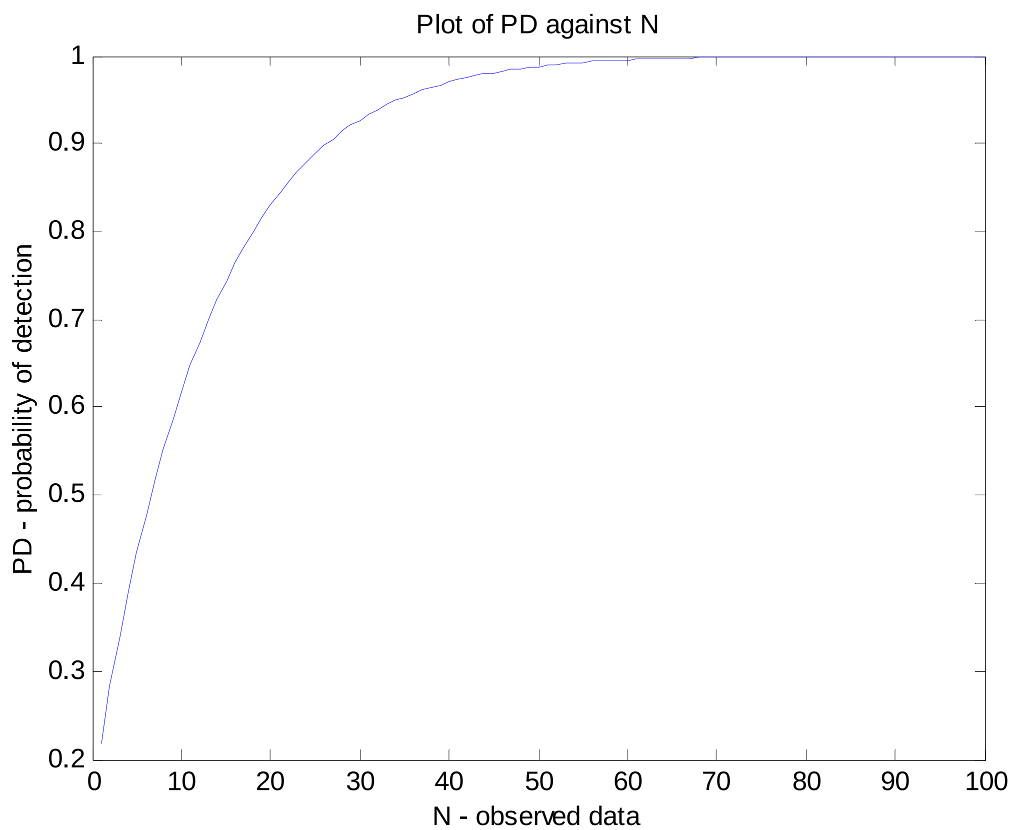

Figure 1

SEQUENTIAL DETECTION

In sequential detection we deal with two thresholds – upper and lower. If the data samples lead to the upper threshold, we choose hypothesis H_1. If the data samples lead to the lower threshold, we choose hypothesis H_0. Otherwise, a data sample is added to reach either threshold. This test can easily be written using equation (A).

The upper threshold is:

$$\sum_{n=0}^{N-1} x[n] \geq 4\ln(A) + N/2$$

$$A\_thresh = 4\ln(A) + N/2$$

The lower threshold is:

$$\sum_{n=0}^{N-1} x[n] \leq 4\ln(B) + N/2$$

$$B\_thresh = 4\ln(B) + N/2$$

Figure 2 shows the plot of both thresholds against 'N' (data samples). This plot is generated using the matlab file "partI_ques02".

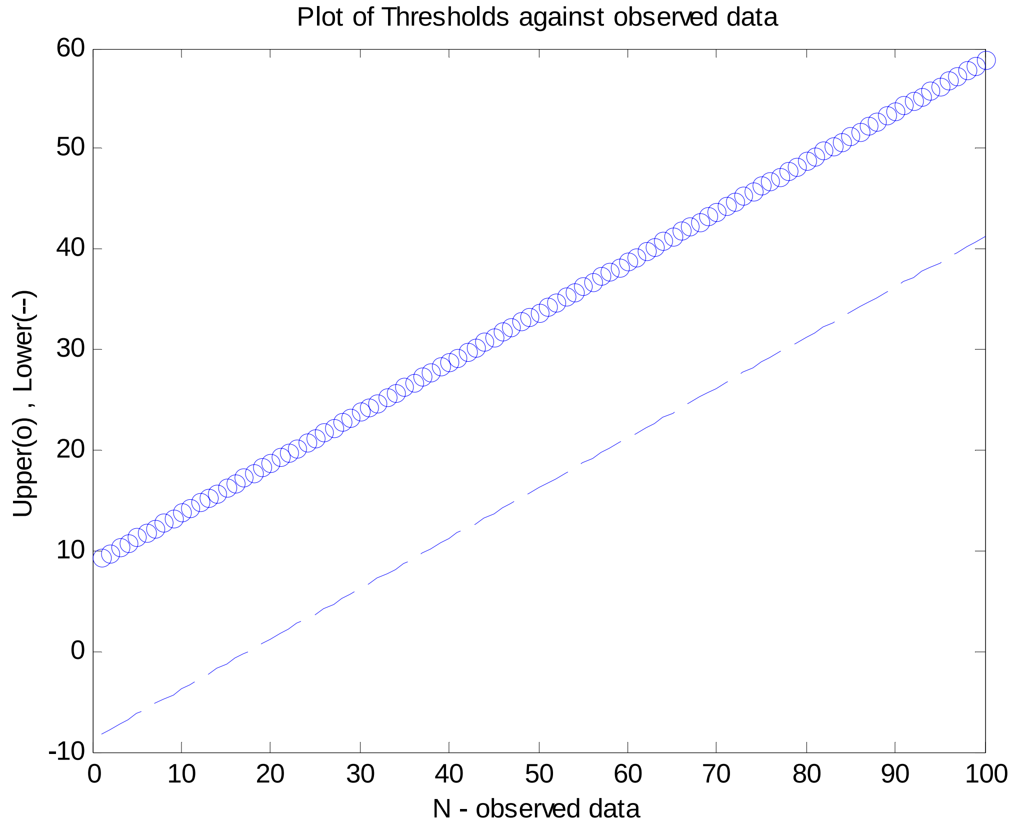

Figure 2

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 equation (A):

$$\sum_{n=0}^{N-1} x[n] \geq 4\ln(\gamma) + N/2$$

$$x[N] \geq 4\ln(\gamma) + N/2 - \sum_{n=0}^{N-1} x[n] + x[N]$$

Description of Matlab Methods for Sequential Test

Matlab methods used for the simulation of part I are included in the appendix. The files used for the purpose are "pfa.m", "pd.m". 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 H_1 is decided. If the test statistic falls below the lower threshold, then 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 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 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.

According to the simulations, eighteen samples are needed on average to get a P_{FA} of 0.1004. Sixteen 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 Figure 1). So we conclude that 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.

PART II

The two hypotheses given to us are

H_0 : x[n] = w[n], n = 0,…..,N-1

H_1 : x[n] = s[n] + w[n], n = 0,…..,N-1

Here 'w[n]' is WGN with zero mean (μ) and variance (σ^2) of 4, and 's[n]' is a sinusoidal signal:

s[n] = Acos(2πn/8)

NEYMAN PEARSON (N-P) LIKELIHOOD RATIO TEST (LRT)

This section gives the mathematical derivation of N-P LRT to choose between H_0 and H_1 using a fixed number of samples N for a given P_{FA}.

$$L(x[n]) = \frac{\exp\left[\dfrac{-1}{2\sigma^2}\sum_{n=0}^{N-1}\left(x[n] - A\cos\left(\dfrac{2\pi n}{8}\right)\right)^2\right]}{\exp\left[\dfrac{-1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n])^2\right]}$$

$$L(x[n]) = \exp\left[\frac{-1}{2\sigma^2}\sum_{n=0}^{N-1}\left(x[n] - \cos\left(\frac{2\pi n}{8}\right)\right)^2 + \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n])^2\right]$$

$$L(x[n]) = \exp\left[\frac{1}{8}\sum_{n=0}^{N-1}\left\{2x[n]\cos\left(\frac{2\pi n}{8}\right) - \left(\cos\left(\frac{2\pi n}{8}\right)\right)^2\right\}\right]$$

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

$$\frac{1}{8}\sum_{n=0}^{N-1}\left\{2x[n]\cos\left(\frac{2\pi n}{8}\right) - \left(\cos\left(\frac{2\pi n}{8}\right)\right)^2\right\} \geq \ln(\gamma)$$

$$2\sum_{n=0}^{N-1} x[n]\cos\left(\frac{2\pi n}{8}\right) - \sum_{n=0}^{N-1}\left(\cos\left(\frac{2\pi n}{8}\right)\right)^2 \geq 8\ln(\gamma)$$

$$\sum_{n=0}^{N-1} x[n]\cos\left(\frac{2\pi n}{8}\right) \geq 4\ln(\gamma) + \frac{1}{2}\sum_{n=0}^{N-1}\left(\cos\left(\frac{2\pi n}{8}\right)\right)^2 \;............(B)$$

Therefore

$$\gamma' = 4\ln(\gamma) + \frac{1}{2}\sum_{n=0}^{N-1}\left(\cos\left(\frac{2\pi n}{8}\right)\right)^2$$

Here γ' is the new threshold. In N-P method, we choose the desired P_{FA} (probability of false-alarm) and accordingly determine the threshold.

SEQUENTIAL DETECTION

In sequential detection we deal with two thresholds – upper and lower. If the data samples lead to the upper threshold, we choose hypothesis H_1. If the data samples lead to the lower threshold, we choose hypothesis H_0. Otherwise, a data sample is added to reach either threshold. This test can easily be written using equation (B).

The upper threshold is:

$$\sum_{n=0}^{N-1} x[n]\cos\left(\frac{2\pi n}{8}\right) \geq 4\ln(A) + \frac{1}{2}\sum_{n=0}^{N-1}\left(\cos\left(\frac{2\pi n}{8}\right)\right)^2$$

$$A\_thresh = 4\ln(A) + \frac{1}{2}\sum_{n=0}^{N-1}\left(\cos\left(\frac{2\pi n}{8}\right)\right)^2$$

The lower threshold is:

$$\sum_{n=0}^{N-1} x[n]\cos\left(\frac{2\pi n}{8}\right) \leq 4\ln(B) + \frac{1}{2}\sum_{n=0}^{N-1}\left(\cos\left(\frac{2\pi n}{8}\right)\right)^2$$

$$B\_thresh = 4\ln(B) + \frac{1}{2}\sum_{n=0}^{N-1}\left(\cos\left(\frac{2\pi n}{8}\right)\right)^2$$

Representing recursively we have:

$$x[N] \geq 4\ln(\gamma) + \frac{1}{2}\sum_{n=0}^{N-1}\left(\cos\left(\frac{2\pi n}{8}\right)\right)^2 - \sum_{n=0}^{N-1} x[n]\cos\left(\frac{2\pi n}{8}\right) + x[N]$$

Description of Matlab Methods for Sequential Test

Matlab methods used for the simulation of this part are included in the appendix. The files used for the purpose are "pfaCos.m", "pdCos.m. 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 H_1 is decided. If the test statistic falls below the lower threshold, then 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 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 H_1. 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_D.

According to the simulations, on average 32 samples are needed to get a P_{FA} of 0.085. On average, 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 sinusoid. The magnitude of the sinusoid 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.

PART III

The two hypotheses given to us are

$H_0 : x[n] = w[n], \qquad n = 0,\ldots,N-1$

$H_1 : x[n] = s[n] + w[n], \qquad n = 0,\ldots,N-1$

Here 'w[n]' is WGN with zero mean (μ) and variance (σ^2) of 1, and 's[n]' is also an white Gaussian sequence with zero mean and variance of one.

NEYMAN PEARSON (N-P) LIKELIHOOD RATIO TEST (LRT)

This section is the mathematical derivation of N-P LRT to choose between H_0 and H_1 using a fixed number of samples 'N' for a given P_{FA}.

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

$$L(x[n]) = \frac{1}{(2)^{N/2}} \frac{\exp\left[\dfrac{-1}{4}\sum_{n=0}^{N-1}(x[n])^2\right]}{\exp\left[\dfrac{-1}{2}\sum_{n=0}^{N-1}(x[n])^2\right]}$$

$$L(x[n]) = \frac{1}{(2)^{N/2}} \exp\left[\frac{-1}{4}\sum_{n=0}^{N-1}(x[n])^2 + \frac{1}{2}\sum_{n=0}^{N-1}(x[n])^2\right]$$

$$L(x[n]) = \frac{1}{(2)^{N/2}} \exp\left[\frac{1}{4}\sum_{n=0}^{N-1}(x[n])^2\right]$$

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

$$\ln\left(\frac{1}{(2)^{N/2}}\right) + \frac{1}{4}\sum_{n=0}^{N-1}(x[n])^2 \geq \ln(\gamma)$$

$$\frac{-N\ln(2)}{2} + \frac{1}{4}\sum_{n=0}^{N-1}(x[n])^2 \geq \ln(\gamma)$$

$$\sum_{n=0}^{N-1}(x[n])^2 \geq 4\ln(\gamma) + 2N\ln(2)\ldots\ldots\ldots\ldots(C)$$

Therefore

$$\gamma' = 4\ln(\gamma) + 2N\ln(2)$$

Here γ' is the new threshold. In N-P method, we choose the desired P_{FA} (probability of false-alarm) and accordingly determine the threshold.

Now

$$P_{FA} = \int_{\gamma'}^{\infty} p(x[n]; H_0)dx[n]$$

and

$$P_D = \int_{\gamma'}^{\infty} p(x[n]; H_1)dx[n]$$

Also

$$E(T(x[n]; H_0) = E\left(\sum_{n=0}^{N-1} (w[n])^2\right)$$

$$E(T(x[n]; H_0) = \sum_{n=0}^{N-1} E\left((w[n])^2\right)$$

$$E(T(x[n]; H_0) = \sum_{n=0}^{N-1} \left(\mathrm{var}(w[n]) + E(w[n])\right)$$

$$E(T(x[n]; H_0) = N$$

and

$$E(T(x[n]; H_1) = E\left(\sum_{n=0}^{N-1} (s[n] + w[n])^2\right)$$

$$E(T(x[n]; H_1) = 2N$$

Also

$$\mathrm{var}(T(x[n]; H_0) = \mathrm{var}\left(\sum_{n=0}^{N-1} (w[n])^2\right)$$

$$\mathrm{var}(T(x[n]; H_0) = 2N$$

and

$$\mathrm{var}(T(x[n]; H_1) = \mathrm{var}\left(\sum_{n=0}^{N-1} (s[n] + w[n])^2\right)$$

$$\mathrm{var}(T(x[n]; H_1) = 4N$$

Therefore

$$P_{FA} = \Pr\{T(x[n]) > \gamma'; H_0\}$$

$$P_{FA} = Q\left(\frac{\gamma' - N}{\sqrt{2N}}\right)$$

$$Q^{-1}(P_{FA}) = \frac{\gamma' - N}{\sqrt{2N}}$$

$$\gamma' = \sqrt{2N}Q^{-1}(P_{FA}) + N$$

and

$$P_D = \Pr\{T(x[n]) > \gamma'; H_1\}$$

$$P_D = Q\left(\frac{\gamma' - 2N}{\sqrt{4N}}\right)$$

$$P_D = Q\left(\frac{\gamma'}{\sqrt{4N}} - \frac{2N}{\sqrt{4N}}\right)$$

$$P_D = Q\left(\frac{\sqrt{2N}Q^{-1}(P_{FA}) + N}{\sqrt{4N}} - \frac{2N}{\sqrt{4N}}\right)$$

$$P_D = Q\left(\frac{\sqrt{2N}Q^{-1}(P_{FA})}{\sqrt{4N}} + \frac{N}{\sqrt{4N}} - \frac{2N}{\sqrt{4N}}\right)$$

$$P_D = Q\left(\frac{Q^{-1}(P_{FA})}{\sqrt{2}} - \frac{\sqrt{N}}{2}\right)$$

The plot of P_D as a function of ‘N’ (data points) is given in Figure 3. This plot was generated from the matlab file “partIII_ques01”.

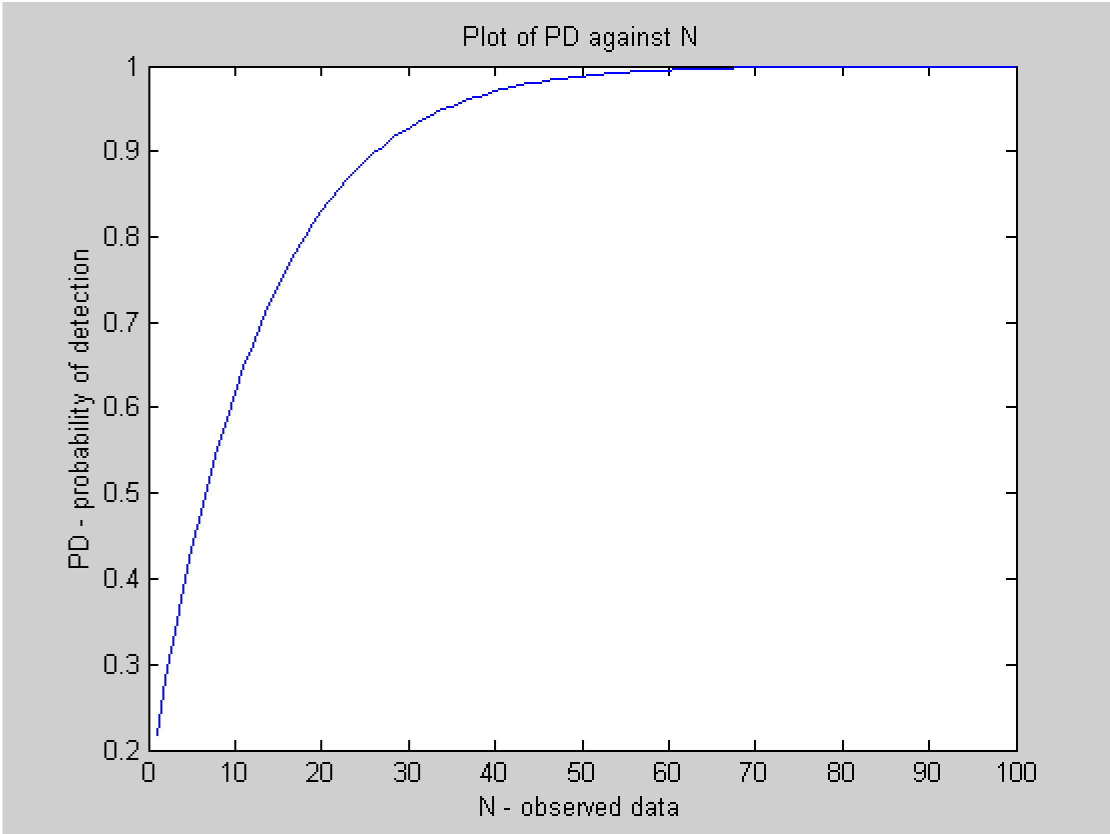

Figure 3

SEQUENTIAL DETECTION

In sequential detection we deal with two thresholds – upper and lower. If the data samples lead to the upper threshold, we choose hypothesis H_1. If the data samples lead to the lower threshold, we choose hypothesis H_0. Otherwise, a data sample is added to reach either threshold. This test can easily be written using equation (B).

The upper threshold is:

$$\sum_{n=0}^{N-1} (x[n])^2 \geq 4\ln(A) + 2N\ln(2)$$

$$A\_thresh = 4\ln(A) + 2N\ln(2)$$

The lower threshold is:

$$\sum_{n=0}^{N-1} (x[n])^2 \leq 4\ln(A) + 2N\ln(2)$$

$$B\_thresh = 4\ln(A) + 2N\ln(2)$$

Representing recursively gives:

$$x(N) \geq 4\ln(\gamma) + 2N\ln(2) - \sum_{n=0}^{N-1}\left(x[n]\right)^2 + x[N]$$

Description of Matlab Methods for Sequential Test

Matlab methods used for the simulation of this part are included in the appendix. The files used for the purpose are "pfaNormal.m", "pdNormal.m". 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 H_1 is decided. If the test statistic falls below the lower threshold, then 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. This is done several hundred times (iterations). With each data sample added to the test, the test statistic is altered. The thresholds remain the same in this case.

For P_{FA}, the data samples are generated using the density under hypothesis 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 H_1. 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_D.

According to the simulations, on average 23 samples are needed to get a P_{FA} of 0.0815. On average, 15 samples are needed to get a P_D of 0.9368. From Figure 3 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.

DESCRIPTION OF PROGRAMS

Q.M

This function is taken from the course book "Fundamentals of Statistical Signal Processing: Detection Theory".

QINV.M

This function is taken from the course book "Fundamentals of Statistical Signal Processing: Detection Theory"

PARTI_QUES01.M

This method plots the probability of detection (P_D) versus the data points (N). The equation for P_D is derived in part I of the project.

PARTI_QUES01.M

This method plots two functions on the same plot. These plots are of upper and lower thresholds versus the data points. The derivation of the upper and lower thresholds is given in part I of the project.

PD.M

This method computes the actual P_D for the first part of the project. The derivations for the thresholds and the test statistic can be seen in part I of the project. An inside "while" loop is executed to keep on adding new data to the test statistic, and update the thresholds until either one of the lower or upper thresholds is reached. An outer "for" loop takes care of the number of trials. The value of P_D is calculated by summing the number of times the test statistic reaches the upper threshold divided by the total number of iterations. "N_avg" takes account of the average number of data samples needed for each trial.

PFA.M

This method computes the actual P_{FA} for the first part of the project. The derivations for the thresholds and the test statistic can be seen in part I of the project. An inside "while" loop is executed to keep on adding new data to the test statistic, and update the thresholds until either one of the lower or upper thresholds is reached. An outer "for" loop takes care of the number of trials. The value of P_{FA} is calculated by summing the number of times the test statistic reaches the upper threshold divided by the total number of iterations. "N_avg" takes account of the average number of data samples needed for each trial. In this case the signal is a DC value.

PDCOS.M

This method computes the actual P_D for the second part of the project. The derivations for the thresholds and the test statistic can be seen in part II of the project. An inside "while" loop is executed to keep on adding new data to the test statistic, and update the thresholds until either one of the lower or upper thresholds is reached. An outer "for" loop takes care of the number of trials. The value of P_{FA} is calculated by summing the number of times the test statistic reaches the upper threshold divided by the total number of iterations. "N_avg" takes account of the average number of data samples needed for each trial. In this case the signal (s[n]), being a function of n, changes constantly and hence changes the mean of the overall signal.

PFACOS.M

This method computes the actual P_{FA} for the second part of the project. The derivations for the thresholds and the test statistic can be seen in part II of the project. An inside "while" loop is executed to keep on adding new data to the test statistic, and update the thresholds until either one of the lower or upper thresholds is reached. An outer "for" loop takes care of the number of trials. The value of P_{FA} is calculated by summing the number of times the test statistic reaches the upper threshold divided by the total number of iterations. "N_avg" takes account of the average number of data samples needed for each trial. In this case the signal (s[n]), being a function of n, changes constantly and hence changes the mean of the overall signal.

PARTIII_QUES01.M

This function plots the P_D versus the data samples. We see from this graph that the number of samples needed for fixed length decision rule is more than the number of data samples needed by sequential techniques.

PDNORMAL.M

This method computes the actual P_D for the third part of the project. The derivations for the thresholds and the test statistic can be seen in part III of the project. An inside "while" loop is executed to keep on adding new data to the test statistic, and update the thresholds until either one of the lower or upper thresholds is reached. An outer "for" loop takes care of the number of trials. The value of P_{FA} is calculated by summing the number of times the test statistic reaches the upper threshold divided by the total number of iterations. "N_avg" takes account of the average number of data samples needed for each trial. In this case both the signal and noise are Gaussian processes making the test statistic and the decision rule very easy to compute.

PFANORMAL.M

This method computes the actual P_{FA} for the third part of the project. The derivations for the thresholds and the test statistic can be seen in part III of the project. An inside "while" loop is executed to keep on adding new data to the test statistic, and update the thresholds until either one of the lower or upper thresholds is reached. An outer "for" loop takes care of the number of trials. The value of P_{FA} is calculated by summing the number of times the test statistic reaches the upper threshold divided by the total number of iterations. "N_avg" takes account of the average number of data samples needed for each trial. In this case both the signal and noise are Gaussian processes making the test statistic and the decision rule very easy to compute.

We see that the functions, "pfa.m", "pfaCos.m", and "pfaNormal.m" are all very similar functions. The main differences are in the way the thresholds are defined. There also exists a difference in the implementation of the test statistic. Similarly "pd.m", "pdCos.m", and "pdNormal.m" are almost identical.