TTT4275 Detection project

Group 5
Student 490105, Marcus Lerfald
Student 490159, Oskar Østby

April 30, 2021

This is a report in the TTK4275 - Estimation, detection and classification course. In the report we will take a look at a detection problem for a cognitive wireless network. In such a network we want to allow secondary users to transmit without affecting the quality of service for primary users. We will model the receiver, try to analyze some performance metrics and in the end develop a working detector that can detect transmit windows in the network for a secondary user. Our detector will in the end be able to detect windows to transmit in with a 1% chance of false detections and very close to 100% rate of detection.

Contents

1 Introduction

This report is a part of an estimation, detection and classification course. In the report we will focus on trying to solve a detection problem involving spectrum detection in a cognitive radio system. In a cognitive radio system we divide users into primary users(PU) and secondary users(SU) to try and maximize the usage of the available spectrum. The PUs in the network requires a certain Quality of Service(QoS) and must be prioritized in the network, but if only the PUs were able to transmit the spectrum would be under utilized. This is where our detection problem appears. If no PU is transmitting the spectrum is free to be utilized, so if we can detect when the spectrum is free we can allow a SU to transmit without affecting the PUs. This is our detection problem and it is quite relevant to our society in which telecommunication and Internet of Things(IoT) devices are rapidly increasing in numbers. They are all sharing the same frequency spectrum, and it is therefore very important to utilize it to the fullest to prevent interference and denial of service. Can we detect when the network is free for a SU without interfering with a PU and without being so pessimistic that the network goes underutilized?

In the network we will analyze orthogonal frequency-division multiplexing(OFDM) is used to modulate data. This method allows many users to efficiently share a bandwidth [3].

In section 2.5 we will go through the basics of detection and, more specifically the Neyman-Pearson detector which is a good detector choice for this problem. After we have established a solid theoretical foundation about detection we will introduce the task we were handed out. In section 4.8 we will show how we have solved all the tasks and derive a solution for the detection problem. Finally we will state some conclusions about our solution in section 5.

2 Theory

In this section we will discuss the different theories that we apply in the rest of the report. To be able to understand our method and implementation we need to define what a detection problem is, what a general detector does and the concept of a Neyman-Pearson detector.

2.1 The general detection problem

In our study, we wanted to detect when a primary unit was transmitting to avoid interference. In an ideal world, we could simply listen to the given frequency and proceed to transmit if we don't hear anything. This is not the case in reality. Due to noise from known and unknown sources, there will always be an observed signal at the given frequency. Some bands have a higher noise floor than others, which must be accounted for. This is the foundation of the detection theory. How do you differentiate the signal of interest from other signals (i.e. noise)? There are several possible ways to formulate such a problem, with different solutions. One of these solutions is the Neyman-Pearson detector.

2.2 Hypothesis testing

Before we explain the Neyman-Pearson detector, we must first visit hypothesis testing. The binary hypothesis test is the basis for our test. This is a way of formulating two different scenarios in a formal manner. First, we consider a null hypothesis. In our case that would be that we only sample background noise, $w[n]$. The alternative hypothesis is that we sample background noise and a signal from the primary unit, $s[n] + w[n]$. We can formulate this hypothesis as

$$\begin{aligned} H_0 &: x[n] = w[n] \\ H_1 &: x[n] = s[n] + w[n] \end{aligned} \tag{1}$$

In our case, $w[n] \sim \mathcal{N}(0, \sigma_w)$ and $s[n] \sim \mathcal{N}(\mu_s, \sigma_s)$. Each of these scenarios has their corresponding probability density function, $p_0(\boldsymbol{x})$ and $p_1(\boldsymbol{x})$, with the mean and variance as stated above. Our goal is to design a detector that can differentiate between these two scenarios based on knowledge about the signal and the given samples, while minimizing the risk of a false alarm [7, p. 38-39].

2.3 Neyman-Pearson detector

The Neyman-Pearson detector is commonly used in binary hypothesis testing when prior information is unknown. The test is formulated as a likelihood ratio test(LRT) as follows [7, p. 47]:

$$L(\mathbf{x}) \triangleq \frac{p_1(\mathbf{x})}{p_0(\mathbf{x})} \begin{cases} \geq \lambda \Rightarrow H_1 \\ < \lambda \Rightarrow H_0 \end{cases} \tag{2}$$

This formulation will minimize a cost function

$$J = (1 - \beta) + \lambda\,(\alpha - \alpha_0) \tag{3}$$

where $\beta = P_D$ is the probability of detection and $\alpha = P_{FA}$ is the probability of false alarm. These are not independent of each other, which is why the detection problem always has to include some sort of trade off. If we aim to detect everything, we must also accept some false alarms. It is therefor common to define an upper limit for the allowed amount of false alarms, α_0. When this is set, we can proceed to deriving the threshold λ. λ is chosen such that $P_{FA}(\lambda) = \alpha_0$. This is in general not trivial, which we will discuss further in section 3.

2.4 Central limit theorem

The central limit theorem is a very useful property from probability theory stating that a sum of N independent random variables with finite mean and variance will converge to a normal distribution as N increases. The distribution will have mean $\mu = \sum_{i=1}^{N} \mu_i$ and variance $\sigma^2 = \sum_{i=1}^{N} \sigma_i$ [4, p. 234].

2.5 Relation between Gamma and χ^2-distributions

Another useful theorem from probability theory is the relationship between the Gamma distribution and the χ^2-distribution. The theorem states that if $X \sim \chi^2(\nu)$, then $cX \sim \Gamma(\nu/2, 2c)$ for $c > 0$ [6].

The gamma distribution also has the nice property that a sum of gamma distributed random variables is also gamma distributed, but with the summed degrees of freedom. See eq. (4).

Given $X_i \sim \Gamma(k_i, \theta)$ and all X_i are independent, then their sum is as follows [6]:

$$\sum_{i=0}^{N} X_i \sim \Gamma\left(\sum_{i=0}^{N} k_i, \theta\right) \tag{4}$$

3 The tasks

We chose the detection task due to the practical advantage it has in telecommunication and IoT. Interference and poor QoS is something we experience in our daily lives and we wanted to further research possible solutions to this problem. In this project we were tasked with creating a proper model of the spectrum we are running our detector on. We will analyze some empirical data and try to see how well our model fits with the data. After we have built our model we will implement a one-sample NP-detector and analyze the statistical properties of the detector, like the probability of detection and false alarm. We will then expand this model to a K-sample detector and then calculate the detection threshold for a generic K-sample detector. Finally we will approximate the performance of this detector and then perform some numerical experiments on data that were handed out.

Our detection problem can be formulated with the following two hypotheses:

$$\begin{aligned} H_0 &: x[n] = w[n], n = 0, 1, \ldots, N-1 \\ H_1 &: x[n] = s[n] + w[n], n = 0, 1, \ldots, N-1 \end{aligned} \tag{5}$$

where $s[n]$ is the waveform of the PU and $w[n]$ is additive white Gaussian noise. In this task we want to build a detector that can detect the H_1 hypothesis, so that a SU can transmit on the channel.

Task 1

In the first task we must develop a model for our signal and noise. We were handed out some samples that we can use to verify that our chosen model fits well with empirical data.

Task 2

In this task, we will consider the special case when we are using a one-sample-detector, i.e. $N = 1$. Then $w[0] \sim \mathcal{N}(0, \sigma_w)$ and $s[0] \sim \mathcal{N}(\mu_s, \sigma_w)$.

We start by formulating the LRT and from there derive the decision rule for when to chose the alternative hypothesis and when to keep our null hypothesis. We want to show that

$$|x[0]|^2 = x_R^2[0] + x_I^2[0] > \lambda'$$

Where x has a real part x_R and an imaginary part x_I.

Task 3

We have been given two data sets, for samples $x[n]$ under H_0 and H_1. We want to show that

$$2x_R^2[0]/\sigma_w^2 + 2x_I^2[0]/\sigma_w^2 \text{ (under } H_0) \text{ and}$$

$$2x_R^2[0]/\left(\sigma_w^2 + \sigma_s^2\right) + 2x_I^2[0]/\left(\sigma_w^2 + \sigma_s^2\right) \text{ (under } H_1)$$

can be modeled as a χ^2-distribution with two degrees of freedom. We use the given data sets to compute σ_s and σ_w and compare them with the χ^2-distribution. We will then derive expressions for P_{FA} and P_D.

Task 4

In this task we generalize the one-sample-detector to include a set of N samples. In this case we are summing up N χ^2-distributed variables with two degrees of freedom, which results in a new χ^2-distribution with 2N degrees of freedom. We will then derive P_D and P_{FA} for the generalized detector and find the optimal threshold λ'.

Task 5

The purpose of this task is to benchmark the strength of our detector by computing the test statistics obtained in section 3. We will then proceed to plot the receiver operating characteristics(ROC) to illustrate the relationship between P_{FA} and P_D.

Task 6

In this task, we will use the Central Limit Theorem to approximate the performance of the Neyman-Pearson detector. We will then plot P_D and P_{FA} as a function of λ' and compare the approximates with the exact solution. This is in general a very useful approach that may prevent difficult calculations as mentioned in section 2.3.

Task 7

In general, P_D will increase with the number of samples, while P_{FA} decreases. In this section we will calculate the relationship between P_D and P_{FA} for a given N and plot the results.

Task 8

In this section we will test our detector on a set of samples and log the amount of detections for two values of P_{FA}. We are granted 100 realizations of a signal at the secondary unit with N = 256. We want to see if the SU is able to detect when a PU is transmitting at each time instant.

4 Implementation and results

4.1 Modeling

A good place to start is to inspect and model the system that you will work on. We need to find two PDFs, one for each hypothesis to be able to create a NP-detector. We are given a dataset of samples from the complex-valued time-domain OFDM signal given by $s[n] = s_r[n] + js_i[n]$. From these samples we can empirically verify that the data is normally distributed with the following PDF:

$$p(s) = p(s_R)\, p(s_I) = \frac{1}{\pi\sigma_s^2} e^{-\frac{1}{\sigma_S^2}|s-\mu_s|^2} \tag{6}$$

We split the sampled signals into their respective real and imaginary part and then we create a plot as shown in fig. 1. This figure is a line plot based on a histogram to make it a bit easier to examine. We can clearly see from the figure that the real and imaginary part of our signal is normally distributed, by the recognizable bell shaped curve.

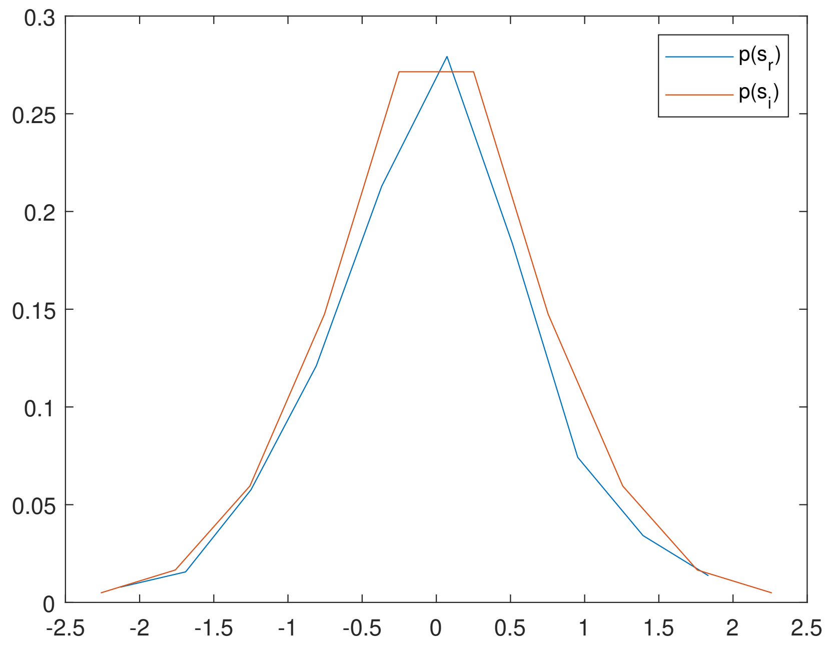

Figure 1: Plot of $s[n]$

We also use the same dataset to compute the empirical expected values of $s_r[n] \cdot s_i[n]$ and $s[n]$ as shown in eq. (7). The source code for computing these values can be found in appendix A.

$$\begin{aligned}\mathbb{E}\{s_r[n]\} &= 0.0313 & \mathbb{E}\{s_i[n]\} &= 6.7252 * 10^{-15} \\ \mathbb{E}\{s[n]\} &= -0.0313 + j6.7252 * 10^{-15} & \mathbb{E}\{s_r[n]s_i[n]\} &= 4.1976 * 10^{-15}\end{aligned} \tag{7}$$

From fig. 1 and our empirical expected values and variances we can say that it seems like a good choice to model the PDF of our signal as a normally distributed Gaussian variable.

4.2 One-sample-detector

In order to formulate the LRT, we need the PDFs for the two scenarios. Under H_0, the mean is 0 and the variance is σ_w^2 as given. Under H_1, $x[n] = s[n] + w[n]$ and $\mathbb{E}(x) = \mu_s + 0$. It also follows that $\mathbb{V}\mathrm{ar}(x) = \sigma_s^2 + \sigma_w^2$. To derive the threshold, we start by manipulating the LRT

$$\frac{p(x; H_1)}{p(x; H_0)} > \lambda$$

$$\implies \frac{\frac{1}{\pi(\sigma_s^2+\sigma_w^2)^{1/2}} \exp(-\frac{1}{2(\sigma_s^2+\sigma_w^2)}|x[0] - \mu_s|^2)}{\frac{1}{\pi(\sigma_w^2)^{1/2}} \exp(-\frac{1}{2\sigma_w^2}|x[0] - 0|^2)} > \lambda$$

Now applying the following:

$$\mu_s = 0, \qquad \frac{1}{\sigma_w^2} - \frac{1}{\sigma_s^2 + \sigma_w^2} > 0$$

$$\implies \ln(\frac{(\sigma_w^2)^{1/2}}{(\sigma_w^2 + \sigma_s^2)^{1/2}}) - \frac{1}{2(\sigma_s^2 + \sigma_w^2)}|x[0] - \mu_s|^2 + \frac{1}{2\sigma_w^2}|x[0]|^2 > \ln(\lambda)$$

$$\implies |x[0]|^2 > \frac{2\sigma_w^2(\sigma_w^2 + \sigma_s^2)}{\sigma_s^2}\left[\ln(\lambda) - \frac{1}{2}\ln(\frac{\sigma_w^2}{\sigma_s^2 + \sigma_w^2})\right] = \lambda'$$

This means that we choose hypothesis H_1 if $|x[0]|^2 > \lambda'$ and H_0 otherwise. In this derivation, we used the assumption that $\mu_s = 0$, which is given. In the last implication, we also used the fact that $\frac{1}{\sigma_w^2} - \frac{1}{\sigma_s^2+\sigma_w^2} > 0$. This is true because σ_w^2 and σ_s^2 are positive, which makes the reciprocal of the sum of variances smaller than the reciprocal of only one of them.

4.3 Performance of the one-sample-detector

In figure 2 we have plotted the histograms of $2|x[0]|^2/\sigma_w^2$ and $2|x[0]|/(\sigma_s^2+\sigma_w^2)$ against the χ^2-distribution with 2 DOF.

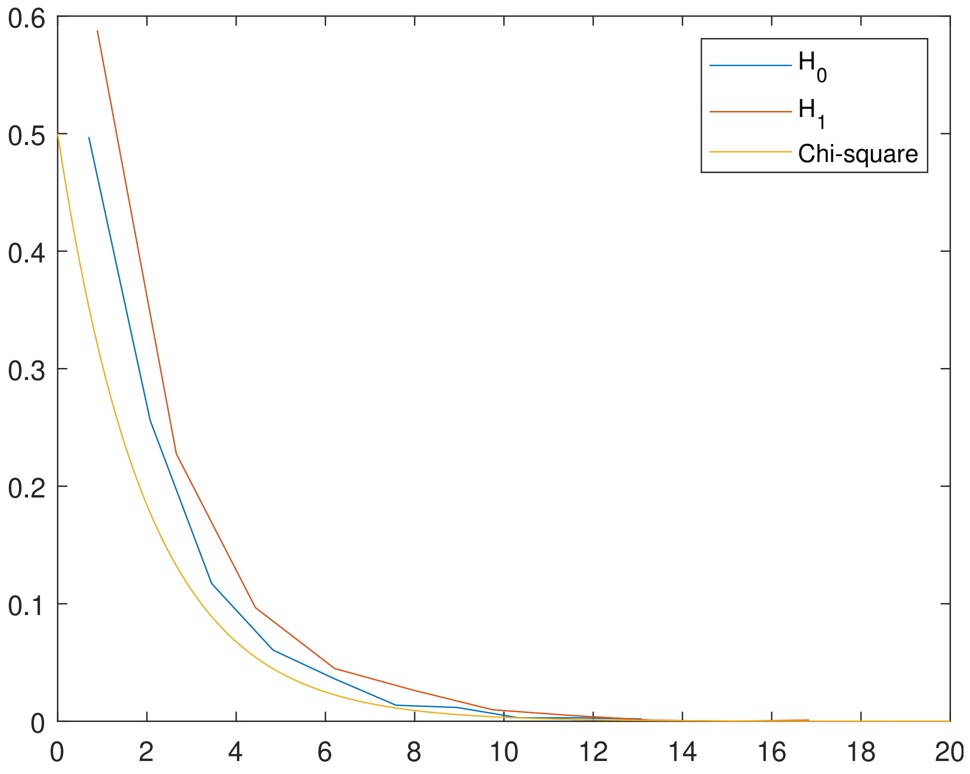

Figure 2: Comparison of empirical data and the χ^2 distribution

We can see that the χ^2-distribution matches quite good, which we in turn can use to calculate P_{FA} and P_D. We start by calculating the probability of false alarm:

$$\begin{aligned} P_{FA} = Prob\left\{|x[0]|^2 > \lambda'; H_0\right\} &= Prob\left\{\frac{2|x[0]|^2}{\sigma_w^2} > \frac{2\lambda'}{\sigma_w^2}; H_0\right\} \\ &= \int_{\frac{2\lambda'}{\sigma_w^2}}^{\infty} \frac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1} \exp(-x/2)\, dx \\ (k=2) &= \int_{\frac{2\lambda'}{\sigma_w^2}}^{\infty} \frac{1}{2} \exp(-x/2)\, dx \\ &= -\exp(-x/2)\Big|_{\frac{2\lambda'}{\sigma_w^2}}^{\infty} = \exp(-\frac{\lambda'}{\sigma_w^2}) \end{aligned} \tag{8}$$

similarly, we get the probability for detection

$$\begin{aligned} P_{FA} = Prob\left\{|x[0]|^2 > \lambda'; H_0\right\} &= Prob\left\{\frac{2|x[0]|^2}{\sigma_s^2+\sigma_w^2} > \frac{2\lambda'}{\sigma_s^2+\sigma_w^2}; H_0\right\} \\ &= \int_{\frac{2\lambda'}{\sigma_s^2+\sigma_w^2}}^{\infty} \frac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1}\exp(-x/2)\,dx \\ (k=2) \implies &= \int_{\frac{2\lambda'}{\sigma_s^2+\sigma_w^2}}^{\infty} \frac{1}{2}\exp(-x/2)\,dx \\ &= -\exp(-x/2)\Big|_{\frac{2\lambda'}{\sigma_s^2+\sigma_w^2}}^{\infty} = \exp(-\frac{\lambda'}{\sigma_s^2+\sigma_w^2}) \end{aligned} \tag{9}$$

An interesting takeaway from this, is that in 9, P_D actually increases with σ_w^2. This is because the detector utilizes the signal power to differentiate between H_0 and H_1. Intuitively, this makes sense as it should be easy to detect the presence of a very noisy signal in an overall less noisy environment.

4.4 Generalized N sample NP-detector

To begin with we define our PDF functions under our two hypothesis' in eq. (10).

$$p_1(\boldsymbol{x}) = p(\boldsymbol{x}; H_1) = \frac{1}{(\pi(\sigma_s^2+\sigma_w^2))^{\frac{N}{2}}}\exp\left(-\frac{1}{2(\sigma_s^2+\sigma_w^2)}\sum_{n=0}^{N-1}|x[n]-\mu_s|^2\right)$$

$$= \frac{1}{(\pi(\sigma_s^2+\sigma_w^2))^{\frac{N}{2}}}\exp\left(-\frac{1}{2(\sigma_s^2+\sigma_w^2)}\sum_{n=0}^{N-1}|x[n]|^2\right) \tag{10a}$$

$$p_0(\boldsymbol{x}) = p(\boldsymbol{x}; H_0) = \frac{1}{(\pi\sigma_w^2)^{\frac{N}{2}}}\exp\left(-\frac{1}{2\sigma_w^2}\sum_{n=0}^{N-1}|x[n]|^2\right) \tag{10b}$$

We insert these expressions into the NP-detector and take the natural logarithm on both sides. We can due this without changing the sign due to $\ln(x)$ being an increasing function. This yields

$$\ln\left(\frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}\right) > \ln(\lambda)$$

$$\ln(p_1(\boldsymbol{x})) - \ln(p_0(\boldsymbol{x})) > \ln(\lambda)$$

We can insert our PDFs and we get the following:

$$-\frac{1}{2(\sigma_s^2+\sigma_w^2)}\sum_{n=0}^{N-1}|x[n]|^2 + \frac{1}{2\sigma_w^2}\sum_{n=0}^{N-1}|x[n]|^2 - \frac{N}{2}\ln(\pi(\sigma_s^2+\sigma_w^2)) + \frac{N}{2}\ln(\pi\sigma_w^2) > \ln(\lambda)$$

We group terms, move all constant terms over to the right side and are left with the following

$$\left(\frac{\sigma_s^2}{2(\sigma_s^2+\sigma_w^2)\sigma_w^2}\right)\sum_{n=0}^{N-1}|x[n]|^2 > \ln(\lambda)+\frac{N}{2}\ln\left(\frac{\sigma_s^2+\sigma_w^2}{\sigma_w^2}\right)$$

Finally, since variances are always positive we can divide both sides by the fraction of variances and we are left with eq. (11) which is our N-sample detection scheme. Since no terms on the left side of the inequality are dependent on our samples we gather them into a single new decision variable λ'.

$$T(\boldsymbol{x})=\sum_{n=0}^{N-1}|x[n]|^2 > \frac{2\sigma_w^2(\sigma_s^2+\sigma_w^2)}{\sigma_s^2}\left[\ln(\lambda)+\frac{N}{2}\ln\left(\frac{\sigma_s^2+\sigma_w^2}{\sigma_w^2}\right)\right]=\lambda' \quad (11)$$

Notice that the general formula with N = 1 gives us the special case discussed in section 4.2. Now from eq. (11) we can find an expression for our P_{FA}. Knowing that $\frac{2}{\sigma^2}|x[n]|^2 \sim \chi_2^2$ then the weighted sum of $|x[n]|^2$, given that all samples are independent, will be $\frac{2}{\sigma^2}T(\boldsymbol{x}) \sim \chi_{2N}^2$ [5]. This leaves us with the probability of a false alarm described in eq. (12).

$$\begin{aligned}P_{FA}=\alpha&=Prob\left\{T(\boldsymbol{x})>\lambda';H_0\right\}\\&=Prob\left\{\frac{2}{\sigma_w^2}T(\boldsymbol{x})>\frac{2\lambda'}{\sigma_w^2};H_0\right\}=1-Q\left(\frac{2\lambda'}{\sigma_w^2},2N\right)\end{aligned} \quad (12)$$

Where Q is the cumulative distribution function(CDF) of the chi square distribution, given by the gamma function and the incomplete gamma function [5]. With this result we get eq. (13) which gives us an expression for λ' which we can use to maximize our detection rate with a given P_{FA}.

$$\alpha=Q\left(\frac{2\lambda'}{\sigma_w^2},2N\right)\implies\lambda'=\frac{\sigma_w^2}{2}Q^{-1}\left(1-\alpha,2N\right) \quad (13)$$

Using the same approach as in eq. (12) we can derive

$$\begin{aligned}P_D=\beta&=Prob\left\{T(\boldsymbol{x})>\lambda';H_1\right\}\\&=Prob\left\{\frac{2}{\sigma_s^2+\sigma_w^2}T(\boldsymbol{x})>\frac{2\lambda'}{\sigma_s^2+\sigma_w^2};H_1\right\}=1-Q\left(\frac{2\lambda'}{\sigma_s^2+\sigma_w^2},2N\right)\end{aligned} \quad (14)$$

which in the end leaves us with the following expression that will be the resulting probability of detection using the NP detector scheme.

$$\beta=1-Q\left(\frac{2\lambda'}{\sigma_s^2+\sigma_w^2},2N\right)=1-Q\left(\frac{2\frac{\sigma_w^2}{2}Q^{-1}\left(1-\alpha,2N\right)}{\sigma_s^2+\sigma_w^2},2N\right) \quad (15)$$

4.5 Performance of a general NP detector

In the previous section, we calculated α and β as functions of λ'. We can now manipulate these equations to express β as a function of α when the DOF are fixed. These receiver operating characteristics illustrate the strength of our detector, as they show how much the probability of detection will decrease if we lower the allowed probability of false alarm. We can utilize the result from eq. (15) to plot the ROC for different DOF. This is shown in fig. 3.

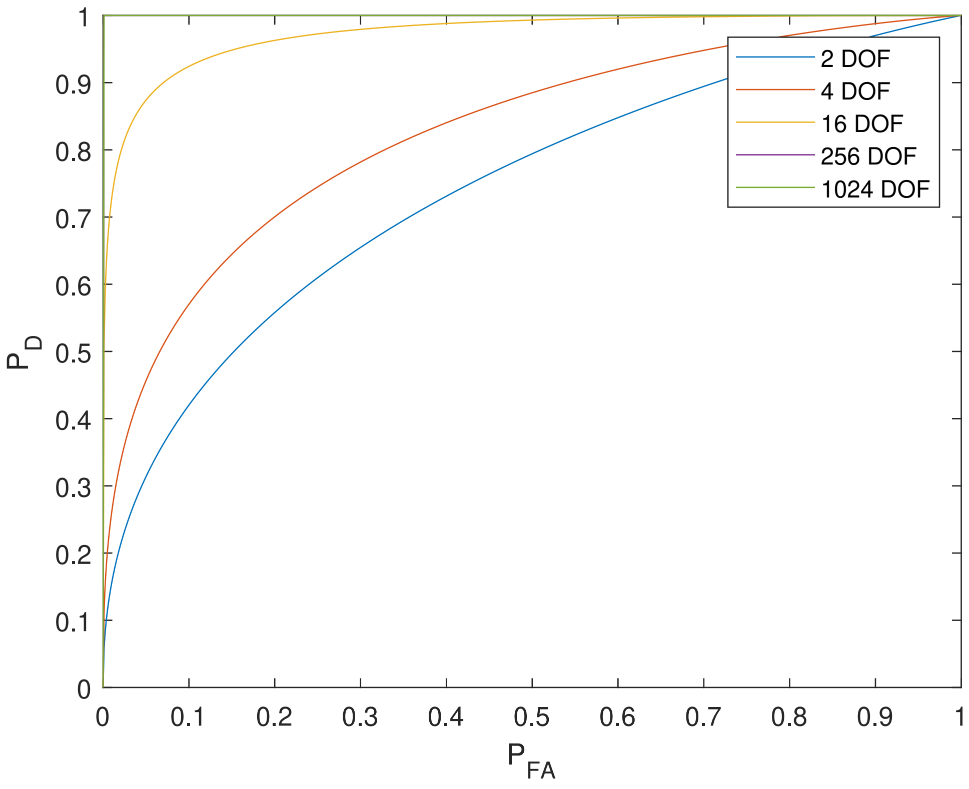

Figure 3: ROC for different DOF

Ideally, this curve should resemble a step function, as that would mean that the probability of detection is very high and we get almost zero false alarms. This is approximately what happens in the case of 256 and 1024 DOF.

Since $\frac{2|x[0]|^2}{\sigma_w^2}$ and $\frac{2|x[0]|^2}{\sigma_w^2+\sigma_s^2}$ are χ^2-distributed with 2N DOF, we can use the scaling theorem from section 2.5 with $c = \frac{\sigma_w^2}{2}$ for H_0 and $c = \frac{\sigma_w^2+\sigma_s}{2}$ for H_1. This will give us a Gamma distribution with scale $2c$ and shape N.

This results in the following PDFs:

$$p(x;H_0) = \frac{1}{\Gamma(N)(\sigma_w^2)^N} x^{N-1} e^{-\frac{x}{\sigma_w^2}}$$
$$p(x;H_1) = \frac{1}{\Gamma(N)(\sigma_w^2+\sigma_s^2)^N} x^{N-1} e^{-\frac{x}{\sigma_w^2 \sigma_s^2}}$$

4.6 Approximating performance of a general NP detector

Our test statistics, $T(\boldsymbol{x})$, which is a sum of random variables can be approximated by the central limit theorem. To use the approximation we must first calculate the expected value of our test statistics under both our hypothesis'.

$$\hat{\mu} = \mathbb{E}\left\{T(\boldsymbol{x})\right. = \sum_{n=0}^{N-1} \mathbb{E}\left\{|x[n]|^2\right\}$$

We apply the transformation of x as we have done earlier

$$\hat{\mu} = \begin{cases} \frac{\sigma_w^2}{2} N\, \mathbb{E}\left\{\frac{2|x[n]|^2}{\sigma_w^2}\right\} & , H_0 \\ \frac{\sigma_w^2+\sigma_s^2}{2} N\, \mathbb{E}\left\{\frac{2|x[n]|^2}{\sigma_w^2+\sigma_s^2}\right\} & , H_1 \end{cases}$$

and using the now known expected value of the transformed version of x we end up with eq. (16) as expressions of the expected value.

$$= \begin{cases} N\sigma_w^2 & , H_0 \\ N(\sigma_w^2+\sigma_s^2) & , H_1 \end{cases} \tag{16}$$

Now we can apply the same method to the variance and we get the following

$$\hat{\sigma}^2 = \mathbb{V}\text{ar}\left\{T(\boldsymbol{x})\right. = \sum_{n=0}^{N-1} \mathbb{V}\text{ar}\left\{|x[n]|^2\right\}$$

$$\hat{\sigma}^2 = \begin{cases} \left(\frac{\sigma_w^2}{2}\right)^2 N\, \mathbb{V}\text{ar}\left\{\frac{2|x[n]|^2}{\sigma_w^2}\right\} & , H_0 \\ \left(\frac{\sigma_w^2+\sigma_s^2}{2}\right)^2 N\, \mathbb{V}\text{ar}\left\{\frac{2|x[n]|^2}{\sigma_w^2+\sigma_s^2}\right\} & , H_1 \end{cases}$$

$$\hat{\sigma}^2 = \begin{cases} N\left(\sigma_w^2\right)^2 & , H_0 \\ N\left(\sigma_w^2+\sigma_s^2\right)^2 & , H_1 \end{cases} \tag{17}$$

Now we can use the results from eqs. (16) and (17) we can use the following PDF as an approximation that is normally distributed.

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} e^{-\frac{1}{2\hat{\sigma}^2}(T(\boldsymbol{x})-\hat{\mu})^2} \tag{18}$$

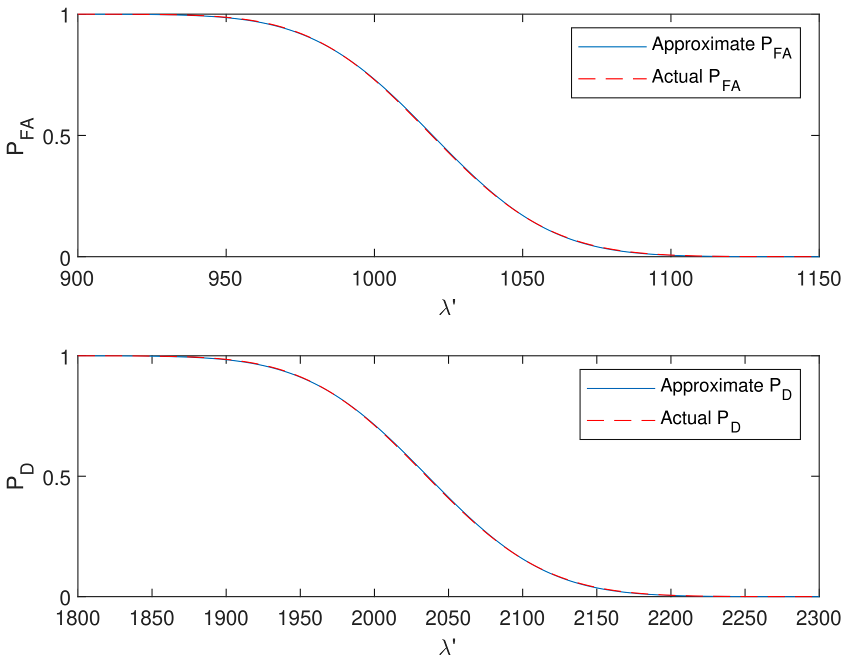

Figure 4: Comparison of approximation of the PDFs

To inspect the validity of the approximation we plot it against the actual PDF in fig. 4. We can see that the approximation is a very good one, which makes sense since the central limit theorem becomes a very good approximation as $N > 30$ and we have $N = 1024$.

4.7 Complexity of our detector

Now using eq. (18) as our PDF we can find a set number of samples that we require to obtain a certain P_{FA} and P_D. So be using the same method as in eqs. (12) and (14) we can derive the following

$$P_{FA} = \alpha = 1 - \Phi\left(\frac{\lambda' - N\sigma_w^2}{\sqrt{N}\sigma_w^2}\right) \implies \lambda' = \sqrt{N}\sigma_w^2\Phi^{-1}(1-\alpha) + N\sigma_w^2 \quad (19)$$

where $\Phi(\cdot)$ and $\Phi^{-1}(\cdot)$ is the regular and inverse CDF of the normal distribution with zero mean and unitary variance.

$$\begin{aligned} P_D = \beta &= 1 - \Phi\left(\frac{\lambda' - N(\sigma_s^2 + \sigma_w^2)}{\sqrt{N}(\sigma_s^2 + \sigma_w^2}\right) \\ &= 1 - \Phi\left(\frac{\sqrt{N}\sigma_w^2\Phi^{-1}(1-\alpha) + N\sigma_w^2 - N(\sigma_s^2 + \sigma_w^2)}{\sqrt{N}(\sigma_s^2 + \sigma_w^2)}\right) \end{aligned}$$

$$\implies \frac{\sqrt{N}\sigma_w^2\Phi^{-1}(1-\alpha) - N\sigma_s^2}{\sqrt{N}(\sigma_s^2 + \sigma_w^2)} = \Phi^{-1}(1-\beta)$$

$$\implies \frac{\Phi^{-1}(1-\alpha)\sigma_w^2 - \sqrt{N}\sigma_s^2}{\sigma_s^2 + \sigma_w^2} = \Phi^{-1}(1-\beta)$$

$$\implies \sqrt{N} = \frac{\Phi^{-1}(1-\alpha)\sigma_w^2 - (\sigma_s^2 + \sigma_w^2)\Phi^{-1}(1-\beta)}{\sigma_s^2}$$

Now since we only want the positive solution of N we can just square both sides and we are left with eq. (20) which relates our desired P_{FA} and P_D with the required number of samples to make a detection.

$$N = \left(\frac{\Phi^{-1}(1-\alpha)\sigma_w^2 - (\sigma_s^2 + \sigma_w^2)\Phi^{-1}(1-\beta)}{\sigma_s^2}\right)^2 \tag{20}$$

4.8 Numerical experiments in PU detection

We are now ready to apply our detector to sampled data. We have been provided a data set of 100 realizations of observed signal, with 256 samples each. At first, we derived our threshold from an acceptable P_{FA} of 0.1. We then reduced our acceptance to 0.01 to illustrate the difference.

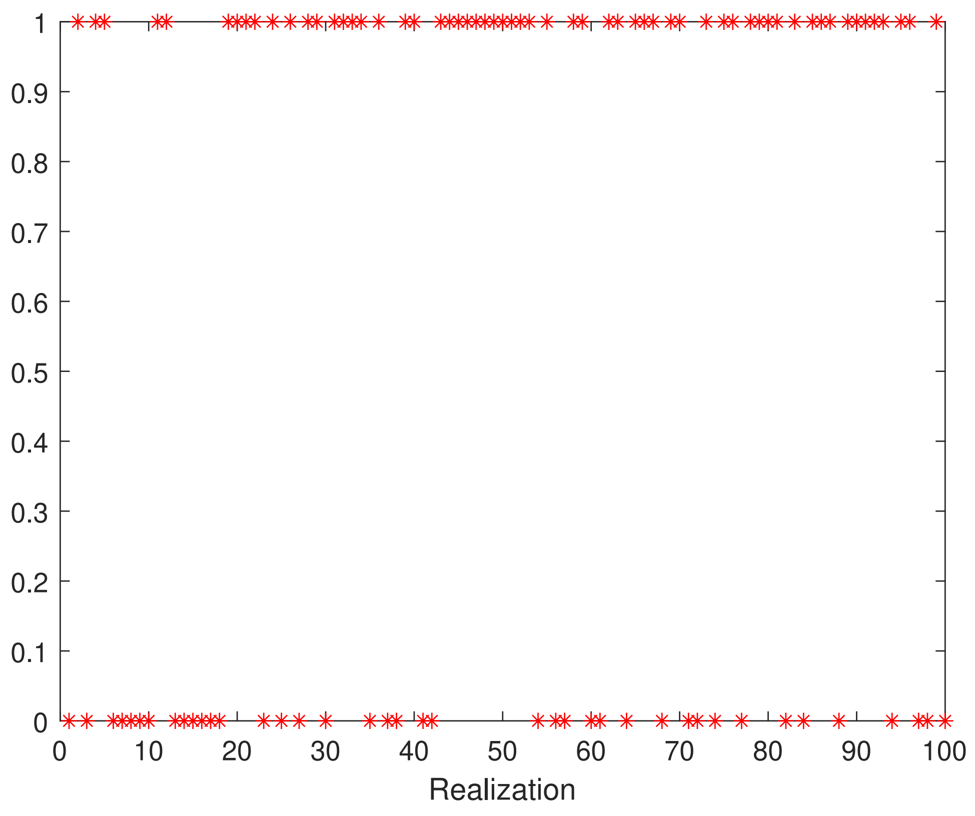

Figure 5: Triggers for $\alpha_0 = 0.1$

In fig. 5, the realizations are listed along the x-axis along with a binary value for hit or miss along the y-axis with a $P_{FA} = 0.1$. Overall, the detector will trigger on a transmitting PU 60 times.

In fig. 6, the acceptable probability of false alarm has been lowered to $P_{FA} = 0.01$, which results in 57 triggers overall. We can observe that the detector will be a bit more conservative when it comes to signaling detections, but the results are still comparable. This is due to the observed characteristics in section 4.5, where 256 DOF results in a very steep ROC curve. Considering that we have 256 samples, we should expect a high P_D even for low P_{FA}. By utilizing eq. (15), we can calculate the P_D for our detector. In both cases, it is approximately 1 down to machine precision. This is very good and it it likely due to our high number of samples and the difference in power between the two signals, making them easier to differentiate.

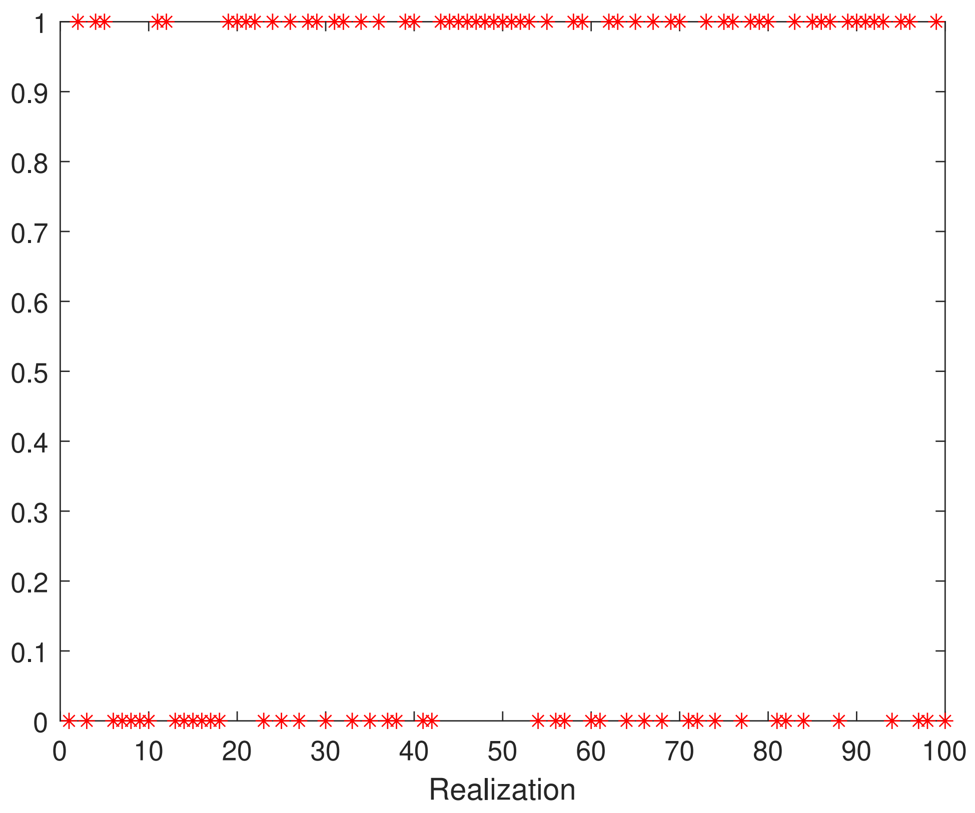

Figure 6: Triggers for $\alpha_0 = 0.01$

If we were to lower our number of samples down to something much lower, like eight, we would get a $P_D = 0.9840$ which is still very good. This is due to $\sigma_s^2 = 5 >> \sigma_w^2 = 1$ leading to a pretty good signal-to-noise ratio. If we were to keep a sample size of 8, but imagine $s[n]$ having $\sigma_s^2 = 1$, our probability of detection would drop to $P_D = 0.7804$.

It is also worth noting that the exact model and the approximation will slightly differ in the number of the detections. In both cases, the approximation utilizing the CLT, will obtain one extra detection. This is the cost of approximation and whether this trade-off can be made or not is application specific.

5 Conclusion

In this report we have researched how to detect transmitting units in a cognitive wireless network. Our study is based on a problem, where several users want to utilize the same spectrum. We want to allow a set of secondary users to use the same spectrum as the primary users without degrading the PU's Quality of Service. We therefore designed a model of our transmitting units to allow further analysis and detector design as explained in section 4.1 and section 4.4. The task at hand was essentially deciding whether a signal was present or not. Due to the binary nature of our problem, we used a Neyman-Pearson detector. In our study, we showed that the the square of the absolute value of our received signal scaled could be modeled as a χ^2-distribution. We could then proceed to derive probability density functions for the two scenarios in our hypothesis to formulate a likelihood ratio test. This was the basis for our detector, now we only had to find our threshold.

In section 4.5, we analyze the the receiver operating characteristics for our detector. This illustrates the relationship between the probability of detection and the probability of false alarm. Since we are using a large sample size of 256, we could settle for a low P_{FA} while still maintaining a high P_D due to the large number of degrees of freedom. In our final test, we used a $\alpha_0 = 0.1$ and $\alpha_0 = 0.01$ to benchmark our detector. We computed our threshold from eq. (11). In our data set of 100 realizations with a sample size of 256, we got 60 detections for $\alpha_0 = 0.1$ and 57 detections for $\alpha_0 = 0.01$. We implemented our findings in MATLAB, in which we were provided data sets for testing. The code for this can be found in appendix A.

In both cases, our probability of detection was approximately 1. We used eq. (15) to calculate our P_D. The reason for the high values of P_D is probably due to a combination of a high sample rate and a high signal to noise ratio as discussed in section 4.8. At this point, we started to experiment with how flexible our detector was. It turns out that we could sample way less and still be able to detect a PU's transmission consistently, even with a lower signal-to-noise ratio. This is of course application dependant, and in some cases, having a $P_D = 0.95$ would not be sufficient. In our scenario, you could save battery on the SU's by only sampling $\frac{1}{10}$th of the time and still maintain a probability of detection of 99.99%.

During this project we have learnt the strength of detection algorithms and their application on real world problems. It has been a very nice introduction to statistical modelling and how to apply detection theory, seeing how the detector performs very satisfactory on the data that was handed out. We have also gained experience in the actual simulation and implementation of detection theory which is very well received.

The open nature of the report structure really is appreciated. From experience very rigid report structures and templates often shifts the focus away from the actual topics that we want to learn over to how academic writing works, which is probably good to know for the future, but it makes the course a lot less enjoyable.

References

[1] H. V. Poor, *An Introduction to Signal Detection and Estimation.* Springer-Verlag, 1988.

[2] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis.* Addison-Wesley Publishing Co., 1990.

[3] M. Ergen, *Mobile Broadband - Including WMAX and LTE*, first. Springer US, 2009, ISBN: 978-0-387-68189-4.

[4] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability & Statistics for Engineers & Scientists*, ninth. Pearson, 2012, ISBN: 978-0-321-62911-1.

[5] *Chi-square distribution*, Apr. 2021. [Online]. Available: `https://en.wikipedia.org/wiki/Chi-square_distribution`.

[6] *Gamma distribution*, Apr. 2021. [Online]. Available: `https://en.wikipedia.org/wiki/Gamma_distribution`.

[7] T. A. Myrvoll, S. Werner, and M. M. Johnsen, "Estimation, detection and classification theory," Course material from TTK4275 at NTNU.

A MATLAB code

Code task 1

```
load_data
clf

%% Generate the s[n] signal.
s = zeros(length(T1_data_Sk_BPSK),1);
N = length(s);
for n=1:N
    for k=1:N
        s(n) = s(n) + T1_data_Sk_BPSK(k)*exp(1i*2*pi*n
            *k/N);
    end
    s(n) = s(n)/sqrt(N);
end

s_r = real(s);
s_i = imag(s);

[p_r, x_r] = hist(s_r);
[p_i, x_i] = hist(s_i);

mu_r = mean(s_r);
mu_i = mean(s_i);
mu_s = mu_r + 1i*mu_i;
E_si_si = mean(s_r.*s_i)

var_r = var(s_r);
var_i = var(s_i);
var_s = var_r + var_i;
sigma_s = sqrt(var_s);

%% Plot pdf(s)

figure(1)
plot(x_r, p_r/sum(p_r));
hold on;
plot(x_i, p_i/sum(p_i));
legend('p(s_r)', 'p(s_i)')

figure(2)
x = -3:.1:3;
```

```
y = normpdf(x, real(mu_s), sigma_s);
plot(x, y);
hold on
[p_g, x_g] = hist(T1_data_Sk_Gaussian);
plot(x_g, p_g/sum(p_g));
legend('p(s)', 'p(s_{gaussian})')
```

Code task 3

```
load_data
clf

%% Generete x with and without s
var_s = var(s_t);
var_w = var(w);
[p_0, x_0] = hist(2*T3_data_x_H0.*conj(T3_data_x_H0)/
    var_w);
[p_1, x_1] = hist(2*T3_data_x_H1.*conj(T3_data_x_H1)/(
    var_s+var_w));

%% Plot pdf(s)
figure(1)
plot(x_0, p_0/sum(p_0));
hold on
plot(x_1, p_1/sum(p_1));
hold on
x = 0:.1:20;
y = chi2pdf(x, 2);
plot(x,y)
legend('x under H_0', 'x under H_1', 'Chi-square')
```

Code task 5

```
load_data
clf

%% Plot ROC (Using chi-square with N DOF)
alpha = 0:0.001:1;
var_w = var(w);
var_s = var(s_t);
figure(1)
for i = [2, 4, 16, 256, 1024]
    beta = gammainc(var_w/(var_w+var_s)* ...
    gammaincinv(alpha,i,'upper'),i,'upper');
    plot(alpha, beta)
    hold on
end
legend('2 DOF', '4 DOF', '16 DOF', '256 DOF', '1024
    DOF')
xlabel("P_{FA}")
ylabel("P_D")
```

Code task 6

```
load_data
clf

%% Approximate test statistic using CLT
lambda = 0:0.1:5000;
var_w = var(w);
var_s = var(s_t);
N = length(s_t);
P_D = 1-normcdf(lambda, N*(var_w+var_s), sqrt(N)*(
    var_w+var_s));
P_FA = 1-normcdf(lambda, N*(var_w), sqrt(N)*(var_w));
P_D_actual = chi2cdf(2*lambda/(var_w+var_s), 2*N, '
    upper');
P_FA_actual = chi2cdf(2*lambda/var_w, 2*N, 'upper');
figure(1)
plot(lambda, P_FA)
hold on
plot(lambda, P_FA_actual)
ylabel('P_{FA}')
xlabel("\lambda'")
legend('Approximate P_{FA}', 'Actual P_{FA}')
figure(2)
plot(lambda, P_D)
hold on
plot(lambda, P_D_actual)
ylabel('P_{D}')
xlabel("\lambda'")
legend('Approximate P_{D}', 'Actual P_{D}')
```

Code task 8

```
load_data
close all

%% Apply NP detector
var_s = 5;
```

```
var_w = 1;
N = 256;
P_FA = 0.1;

lambda = sqrt(N)*var_w*norminv(1 - P_FA)+N*var_w;
lambda_true = (var_w/2)*chi2inv(1 - P_FA, 2*N);

NP_detector = NaN(100,1);
hits = 0;
for i = 1:100
    NP_detector(i,1) = sum(abs(T8_numerical_experiment
        (1:N,i)).^2) > lambda;
    if NP_detector(i,1) == 1
        hits = hits + 1;
    end
end
plot(1:100, NP_detector(:,1),'r*')
```