

Spectrum Sensing in OFDM Cognitive Radios

Markus Søvik Gunnarsson, Floris Lutz

April 30, 2020



Title image: [6]

Optimising existing cellular network infrastructure and making high speed wireless data transfers even cheaper and easily available, so-called *cognitive radio* systems have evolved the last years. Their functionality highly relies on precise transmitting detection, checking whether a specific bandwidth is used for the moment or not. This study aims to develop such a transmitting detection by using a Neyman-Pearson detector. Besides the mathematical derivation of this decoder, it also contains an implementation of the decoder itself in Python and an evaluation using test data.

The implemented detector works quite well on the used data, especially if test statistics can be made from a bigger amount of samples. Conclusive a Newman-Pearson-Detector fulfils its purpose on transmitting detection in cognitive radios.

Contents

1	Introduction	4
2	Theory	5
2.1	Stochastic random processes	5
2.2	Normal distribution	6
2.3	Chi-squared-distribution	7
2.4	Central limit theorem	8
2.5	General detection problem and Neyman Pearson detector	9
3	Task	10
3.1	Model building	10
3.2	One-sample detector	11
3.3	Performance of the one-sample detector	12
3.4	Neyman-Person detector with a data set of N samples.	14
3.5	Performance of a general NP-detector	15
3.6	Approximate performance of a general NP detector	18
3.7	Complexity of the detector	20
3.8	Numerical experiments in PU detection	21
4	Implementation and results	22
4.1	Programming language	22
4.2	Used libraries	22
4.3	Frequently used functions	22
4.4	Results	24
4.5	Numerical experiment results	29
5	Conclusion	31
	References	32

1 Introduction

Since there is an ever growing demand for high data rates, wireless communication engineers and companies around the world are continuously updating their wireless communication standards. By now every device in common 3G/4G-networks uses a fixed part of the frequency spectrum. As for these earlier approaches, a big part of the network's high speed bandwidths is reserved for system-relevant and extra-paying users, also referred as primary users. For the first time it now is possible to dispose this extra bandwidths to common users (secondary users) as far as they are not requested by primary users. This new approach is called *cognitive radio* and will only work if secondary users have a good enough detector to sense if a channel on the frequency spectrum is being used or not. [1]

The primary goal for this project is to derive and implement such a detector by using the theory and knowledge acquired in the course TTT4275 Estimation, detection and classification. By looking at this practical example (spectrum Sensing in OFDM Cognitive Radios) and applying the things we have learnt throughout the course, we hope to increase our knowledge on statistics, detection theory, scripting/programming and plotting/graphing.

This report is divided in primarily four different section; Theory, task, implementation/result and conclusion.

In theory we are discussing the most relevant background information required to understand the task, which includes; Normal/Chi-squared distribution, the central limit theorem and Neyman-Pearson detectors.

The section "Task" gives a description and solution to design a detector, from setting up a model to performing a numerical experiment, divided into eight subtasks.

In implementation and result we will take a look at the programming language used along with the most frequently used libraries and function. This section also includes tables and figures from the practical aspect of some of the tasks.

At last we have the conclusion section where the key points from the task and results will be mentioned. To wrap things up there will be some personal thoughts of what has been learned from this project.

2 Theory

2.1 Stochastic random processes

Since a secondary user doesn't know the message transmitted by a primary user, he has no knowledge about the exact shape of the signal he is supposed to detect. In addition to this signal all real sensors measure some background noise which exact shape isn't known either.

Therefore a reliable model describing our problem can't be built from **deterministic** signals, instead we have to model both signal and noise as **stochastic** processes, which don't make any prediction about the signal's exact shape but its **statistical probabilities**.

"A stochastic process is a family or ensemble of signals corresponding to every possible outcome of a certain signal measurement or experiment. Each signal in the ensemble is called a "realisation" of the process." [2]

So instead of defining a signal as a deterministic sequence like

$$x[n] = \sin(\omega n) \quad (1)$$

we define it as a random signal with known statistical properties:

$$x[n] \sim \mathcal{N}(\mu, \sigma^2) \quad (2)$$

In this example, x is a normally distributed random variable with (true) mean μ and variance σ^2 .

Each random variable with known statistical properties can be described by a **Probability density function** (PDF). This function $p(x)$ is a continuous function that links every possible outcome of the random variable with its likelihood. Since the likelihoods of all possible outcomes have to add up to 100%, the integral of the PDF over all possible outcomes of x has to be one.

$$\int_{-\infty}^{\infty} p(x)dx = 1 \quad (3)$$

If the integral's borders are some explicit values a and b , the corresponding integral of the PDF results in the probability of the outcome of x being inside the interval $[a, b]$.

$$\int_a^b p(x)dx = P(a \leq x \leq b) \quad (4)$$

The indefinite integral of the PDF is also called the **Cumulative density function** (CDF) and is always in the range between 0 and 1.

In the following chapters two of the most common probability distributions are explained.

2.2 Normal distribution

Supposed, a process x is normally or Gaussian distributed

$$x[n] \sim \mathcal{N}(\mu, \sigma^2) \quad (5)$$

its statistical properties can be described by the following PDF

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \quad (6)$$

and CDF

$$P(x) = \frac{1}{2} \left(1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right) \right) \quad (7)$$

with $\operatorname{erf}()$ meaning the error function. A plot of the PDF for a normal distributed $p(x)$ is shown in figure 1. [5]

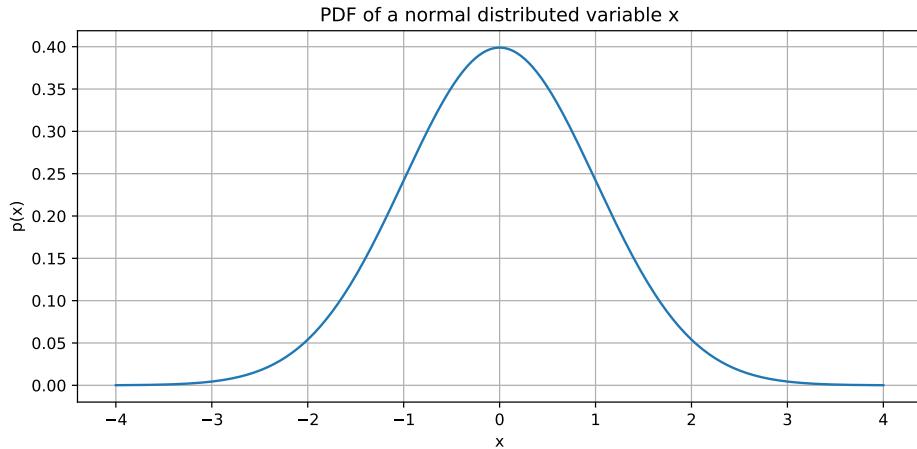


Figure 1: PDF of a normal distributed variable with $\mu = 0$ and $\sigma^2 = 1$.

The integral (also generally known as the tail-/survival function)

$$\int_{\lambda}^{\infty} p(x)dx = [P(x)]_{\lambda}^{\infty} \quad (8)$$

is for the Gaussian case often referred as the so-called Q-function.

$$\int_{\lambda}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} dx = Q\left(\frac{\lambda-\mu}{\sigma}\right) \quad (9)$$

While the classical normal distribution only works for real outcomes ($x \in \mathbb{R}$), there also is a complex version ($x \in \mathbb{C}$) [1] with the PDF

$$p(x) = p(x_r)p(x_i) = \frac{1}{\pi\sigma^2} \cdot \exp\left\{-\frac{|x - \mu|^2}{\sigma^2}\right\} \quad (10)$$

where

$$x[x] = x_r[n] + jx_i[n] \text{ with } x_r, x_i \sim \mathcal{N}(\mu, \frac{\sigma^2}{2}) \quad (11)$$

A plot of the PDF for a complex normal distributed $p(x)$ is shown in figure 2.

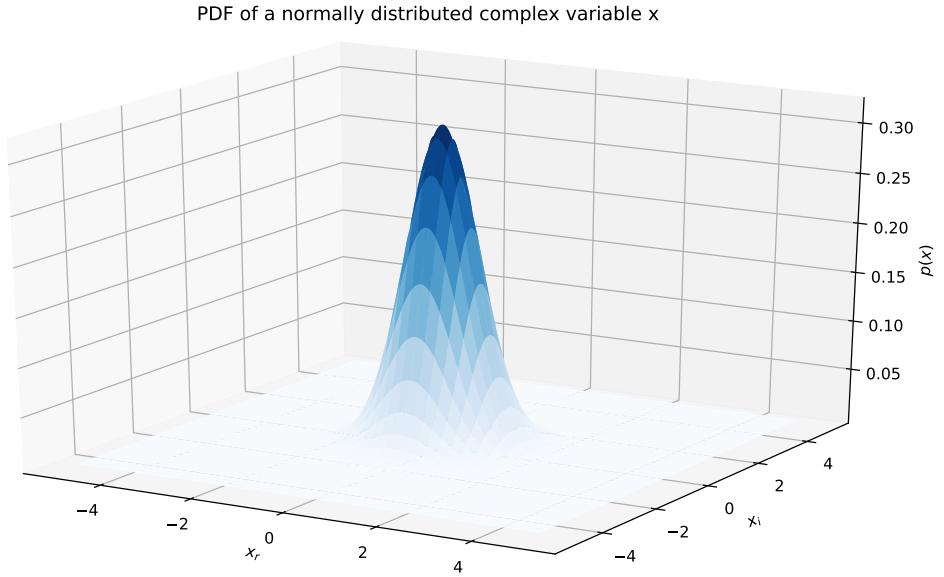


Figure 2: PDF of a normal distributed complex variable with $\mu = 0$ and $\sigma^2 = 1$.

2.3 Chi-squared-distribution

If several realisations of Gaussian random variables are squared and afterwards added up

$$z[n] = \sum_{l=0}^k x_l^2[n] \text{ with } x \sim \mathcal{N}(0, 1) \quad (12)$$

then z is a chi-squared variable with k degrees of freedom.

$$x[n] \sim \chi_k^2 \quad (13)$$

The PDF for a chi-squared random variable z with k degrees of freedom is

$$p(z) = \frac{1}{2^{\frac{k}{2}} (\frac{k}{2} - 1)!} \cdot z^{\frac{k}{2} - 1} \cdot \exp\left\{-\frac{z}{2}\right\} \quad (14)$$

while the CDF can be described as

$$p(z) = \frac{1}{(\frac{k}{2} - 1)!} \cdot \gamma\left(\frac{k}{2}, \frac{z}{2}\right) \quad (15)$$

with γ being the lower incomplete gamma function. A plot of the PDF for $p(z)$ with different degrees of freedom is shown in figure 3. [5]

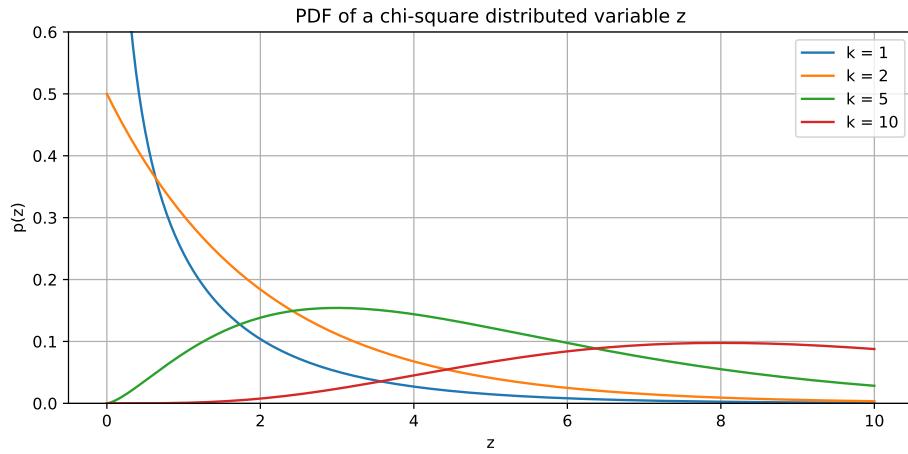


Figure 3: PDF of a chi-square distributed variable z with different degrees of freedom

2.4 Central limit theorem

"The central limit theorem states that for large n the random variable

$$X = \sum_{i=1}^n X_i \quad (16)$$

where X_i is a random variable with any distribution but with finite mean μ_i and variance σ_i^2 , can be approximated as a Gaussian random variable with

$$\text{mean } \mu = \sum_{i=1}^n \mu_i \text{ and variance } \sigma^2 = \sum_{i=1}^n \sigma_i^2 . "[1] \quad (17)$$

Chi-square distributions with a high number of degrees of freedom can for instance be approximated with normal distributions.

2.5 General detection problem and Neyman Pearson detector

Detection problems occur when the goal is to detect "*a rare event s based on a noisy observation x.*" [3] For the composition of x there are two **hypotheses**:

$$H_0 : x[n] = w[n], \quad n = 0, 1, \dots N - 1 \quad (18)$$

$$H_1 : x[n] = w[n] + s[n], \quad n = 0, 1, \dots N - 1 \quad (19)$$

for which the statistical properties of the noise w and the signal s are usually known, s might even sometimes be a deterministic signal. In order to work well, a detector has a **decision rule** that is supposed to fit the detector's **decision regions** Ω_1 and Ω_0 to the actual hypotheses H_1 and H_0 the best way possible. With two hypotheses and two decision regions, each decision rule has four possible outcomes listed in figure 4.

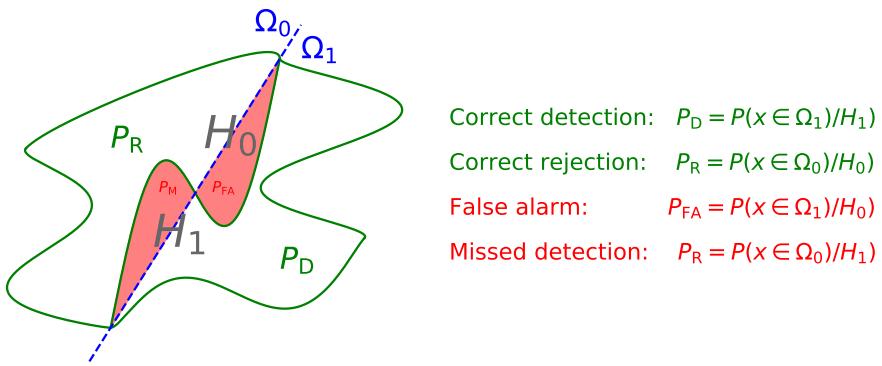


Figure 4: Possible outcomes of a decision rule.[3]

For the most cases, the decision rule relies on a **Likelihood ratio test** (LRT), in which the ratio of the two PDFs for different hypotheses is compared to a threshold λ .

$$\frac{p(x/H_1)}{p(x/H_0)} \begin{cases} \geq \lambda & \Rightarrow H_1 \\ < \lambda & \Rightarrow H_0 \end{cases} \quad (20)$$

While the ratio between the two PDFs (also referred to as **test statistics** $T(x)$) is dependent on the problem, the threshold is dependent on the used method. [3] The solution presented in this report uses a **Neyman-Pearson-Detector**, in which the threshold is determined from a false alarm probability the designer can choose.

The lower this false-alarm probability is chosen, the lower is also the corresponding detection probability, so finding the right value for P_{FA} is always a trade-off. The relation between P_{FA} and P_D is called the **Receiver operating characteristics** (ROC). [4]

3 Task

3.1 Model building

In order to find the PDFs, $p(x/H_0)$ and $p(x/H_1)$ associated with the respective model hypothesis, we take a look at the signal $x[n]$ for the complex-valued time-domain OFDM signal sequence $s[n]$. The signal $s[n]$ is generated by inverse Fourier transforming a sequence $S[k]$:

$$s[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S[k] \cdot \exp\left\{-\frac{j2\pi nk}{N}\right\}, n = 0, 1, \dots, N-1 \quad (21)$$

If it's supposed that $s_r \sim \mathcal{N}(\mu_{sr}, \frac{\sigma_s^2}{2})$ and $s_i \sim \mathcal{N}(\mu_{si}, \frac{\sigma_s^2}{2})$, it follows for a single sample of $s[n]$ that:

$$p(s) = p(s_r)p(s_i) = \frac{1}{\sqrt{2\pi\frac{\sigma_s^2}{2}}} \cdot \exp\left\{-\frac{(s_r - \mu_{sr})^2}{2\frac{\sigma_s^2}{2}}\right\} \cdot \frac{1}{\sqrt{2\pi\frac{\sigma_s^2}{2}}} \cdot \exp\left\{-\frac{(s_i - \mu_{si})^2}{2\frac{\sigma_s^2}{2}}\right\} \quad (22)$$

$$= \frac{1}{\pi\sigma_s^2} \cdot \exp\left\{-\frac{(s_r - \mu_{sr})^2 + (s_i - \mu_{si})^2}{\sigma_s^2}\right\} \quad (23)$$

$$= \frac{1}{\pi\sigma_s^2} \cdot \exp\left\{-\frac{|s_r + js_i - \mu_{sr} - j\mu_{si}|^2}{\sigma_s^2}\right\} \quad (24)$$

$$= \frac{1}{\pi\sigma_s^2} \cdot \exp\left\{-\frac{|s - \mu_s|^2}{\sigma_s^2}\right\} \quad (25)$$

where you can see from equation (25) that $p(s)$ can be modelled as function with a complex Gaussian distribution and variance σ_s^2 . We also model the noise $w[n]$ with a similar distribution and a variance σ_w^2 .

If we assume that the (true) means are $\mu_w = \mu_s = 0$, the PDFs for the two hypothesis turn out to be:

$$p(x/H_0) = p(w) = \frac{1}{\pi\sigma_w^2} \exp\left\{-\frac{|x|^2}{\sigma_w^2}\right\} \quad (26)$$

$$p(x/H_1) = P(w + s) = \frac{1}{\pi\sigma_w^2 + \sigma_s^2} \exp\left\{-\frac{|x|^2}{\sigma_w^2 + \sigma_s^2}\right\} \quad (27)$$

3.2 One-sample detector

A one-sample detector can be a primitive and easy way of solving the detection problem. With a single sample $w[0]$, that is a zero-mean complex Gaussian random variable with variance σ_w^2 , we can use the knowledge of σ_s^2 and σ_w^2 to derive a one-sample Neyman-Person detector.

As described in section 2.5 a NP detector is given by the following equation

$$\frac{p(x/H_1)}{p(x/H_0)} \begin{cases} \geq \lambda & \Rightarrow H_1 \\ < \lambda & \Rightarrow H_0 \end{cases} \quad (28)$$

when inserted with our probability density functions $p(x/H_0)$ and $p(x/H_1)$ equals

$$\frac{\frac{1}{\pi\sigma_w^2+\sigma_s^2} \exp\left\{-\frac{|x|^2}{\sigma_w^2+\sigma_s^2}\right\}}{\frac{1}{\pi\sigma_s^2} \exp\left\{-\frac{|x|^2}{\sigma_w^2}\right\}} \begin{cases} \geq \lambda & \Rightarrow H_1 \\ < \lambda & \Rightarrow H_0 \end{cases} \quad (29)$$

leading to

$$\frac{\sigma_w^2}{(\sigma_w^2 + \sigma_s^2)} \exp\left\{|x|^2\left(\frac{1}{\sigma_w^2} - \frac{1}{\sigma_w^2 + \sigma_s^2}\right)\right\} \begin{cases} \geq \lambda & \Rightarrow H_1 \\ < \lambda & \Rightarrow H_0 \end{cases} \quad (30)$$

if we take the natural logarithm of that, we end up with

$$\ln\left(\frac{\sigma_w^2}{(\sigma_w^2 + \sigma_s^2)}\right) + |x|^2\left(\frac{1}{\sigma_w^2} - \frac{1}{\sigma_w^2 + \sigma_s^2}\right) \begin{cases} \geq \ln(\lambda) & \Rightarrow H_1 \\ < \ln(\lambda) & \Rightarrow H_0 \end{cases} \quad (31)$$

and

$$|x|^2\left(\frac{1}{\sigma_w^2} - \frac{1}{\sigma_w^2 + \sigma_s^2}\right) \begin{cases} \geq \ln(\lambda) - \ln\left(\frac{\sigma_w^2}{(\sigma_w^2 + \sigma_s^2)}\right) & \Rightarrow H_1 \\ < \ln(\lambda) - \ln\left(\frac{\sigma_w^2}{(\sigma_w^2 + \sigma_s^2)}\right) & \Rightarrow H_0 \end{cases} \quad (32)$$

which leads to the detector's test statistics $T(x)$ being given as

$$T(x) = |x|^2 \begin{cases} \geq \lambda' & \Rightarrow H_1 \\ < \lambda' & \Rightarrow H_0 \end{cases} \quad (33)$$

where λ' is

$$\lambda' = \frac{\ln(\lambda) - \ln\left(\frac{\sigma_w^2}{(\sigma_w^2 + \sigma_s^2)}\right)}{\frac{1}{\sigma_w^2} - \frac{1}{\sigma_w^2 + \sigma_s^2}} \quad (34)$$

3.3 Performance of the one-sample detector

To check the performance of the one-sample detector we found in section 3.2, we verify that the detector can be modelled with a Chi-squared-distribution and then compute the probability of detection P_D and probability of false alarm P_{FA} .

Since the test statistics $T(x)$ is written as

$$T(x) = |x[n]|^2 = x_r^2[n] + x_i^2[n] \quad (35)$$

with x_r and x_i normal distributed as $\mathcal{N}\left(0, \frac{\sigma^2}{2}\right)$, a slightly adjusted statistics is

$$T\left(\frac{2x}{\sigma^2}\right) = \left(\frac{2x_r^2}{\sigma^2}\right) + \left(\frac{2x_i^2}{\sigma^2}\right) = \chi_2^2 \quad (36)$$

Using this adjusted chi-squared model for our two hypotheses H_0 and H_1 , we will end up with

$$p(T/H_0) = \frac{2}{\sigma_w^2} \cdot \frac{1}{2^{\frac{k}{2}} \Gamma(\frac{k}{2})} \cdot \left(\frac{2}{\sigma_w^2} T\right)^{\frac{k}{2}-1} \cdot \exp\left\{-\frac{1}{2} \cdot \frac{2}{\sigma_w^2} T\right\} \quad (37)$$

$$p(T/H_1) = \frac{2}{\sigma_w^2 + \sigma_s^2} \cdot \frac{1}{2^{\frac{k}{2}} \Gamma(\frac{k}{2})} \cdot \left(\frac{2}{\sigma_w^2 + \sigma_s^2} T\right)^{\frac{k}{2}-1} \cdot \exp\left\{-\frac{1}{2} \cdot \frac{2}{\sigma_w^2 + \sigma_s^2} T\right\} \quad (38)$$

and $k = 2$ degrees of freedom as $T(x)$ is the sum of two squared Gaussian variables. (See also section 2.3). This makes the gamma function $\Gamma(1) = 1$ and results in

$$p(T/H_0) = \frac{2}{\sigma_w^2} \cdot \frac{1}{2} \cdot \exp\left\{-\frac{T}{\sigma_w^2}\right\} = \frac{1}{\sigma_w^2} \cdot \exp\left\{-\frac{T}{\sigma_w^2}\right\} \quad (39)$$

$$p(T/H_1) = \frac{2}{\sigma_w^2 + \sigma_s^2} \cdot \frac{1}{2} \exp\left\{-\frac{T}{\sigma_w^2 + \sigma_s^2}\right\} = \frac{1}{\sigma_w^2 + \sigma_s^2} \cdot \exp\left\{-\frac{T}{\sigma_w^2 + \sigma_s^2}\right\} \quad (40)$$

The probability of false alarm is given by

$$P_{FA} = \int_{\lambda'}^{\infty} p(T|H_0) dT = \int_{\lambda'}^{\infty} \frac{1}{\sigma_w^2} \cdot e^{-\left(\frac{T}{\sigma_w^2}\right)} dT \quad (41)$$

leading to

$$\left[-e^{-\left(\frac{T}{\sigma_w^2}\right)} \right]_{\lambda'}^{\infty} \quad (42)$$

and evaluated from λ' to ∞

$$P_{FA} = e^{-\left(\frac{\lambda'}{\sigma_w^2}\right)} \quad (43)$$

The same can be done for the probability of detection, which is given by

$$P_D = \int_{\lambda'}^{\infty} p(T|H_1) dT = \int_{\lambda'}^{\infty} \frac{1}{\sigma_w^2 + \sigma_s^2} \cdot e^{-\frac{T}{\sigma_w^2 + \sigma_s^2}} dT \quad (44)$$

again is leading to

$$\left[-e^{-(\frac{T}{\sigma_w^2 + \sigma_s^2})} \right]_{\lambda'}^{\infty} \quad (45)$$

and at last evaluated from λ' to ∞

$$P_D = e^{-(\frac{\lambda'}{\sigma_w^2 + \sigma_s^2})} \quad (46)$$

3.4 Neyman-Person detector with a data set of N samples.

To further improve the detector we can now compute a Neyman-Person detector with multiple samples N and the associate threshold λ' .

The computation is quite similar to the computation for the one-sample detector in section 3.2, while the main difference is the newly added product of the single sample probability distributions $p(x/H_1)$ and $p(x/H_0)$.

$$\frac{\prod_{n=0}^{N-1} \frac{1}{\pi\sigma_w^2 + \sigma_s^2} \cdot \exp\left\{-\frac{|x|^2}{\sigma_w^2 + \sigma_s^2}\right\}}{\prod_{n=0}^{N-1} \frac{1}{\pi\sigma_s^2} \cdot \exp\left\{-\frac{|x|^2}{\sigma_w^2}\right\}} \begin{cases} \geq \lambda & \Rightarrow H_1 \\ < \lambda & \Rightarrow H_0 \end{cases} \quad (47)$$

leading to

$$\prod_{n=0}^{N-1} \frac{\sigma_w^2}{(\sigma_w^2 + \sigma_s^2)} \cdot \exp\left\{|x|^2\left(\frac{1}{\sigma_w^2} - \frac{1}{\sigma_w^2 + \sigma_s^2}\right)\right\} \begin{cases} \geq \lambda & \Rightarrow H_1 \\ < \lambda & \Rightarrow H_0 \end{cases} \quad (48)$$

and

$$\left(\frac{\sigma_w^2}{(\sigma_w^2 + \sigma_s^2)}\right)^N \cdot \exp\left\{\sum_{n=0}^{N-1} |x|^2\left(\frac{1}{\sigma_w^2} - \frac{1}{\sigma_w^2 + \sigma_s^2}\right)\right\} \begin{cases} \geq \lambda & \Rightarrow H_1 \\ < \lambda & \Rightarrow H_0 \end{cases} \quad (49)$$

if we take the natural logarithm of that, we end up with

$$N \cdot \ln\left(\frac{\sigma_w^2}{(\sigma_w^2 + \sigma_s^2)}\right) + \left(\frac{1}{\sigma_w^2} - \frac{1}{\sigma_w^2 + \sigma_s^2}\right) \sum_{n=0}^{N-1} |x|^2 \begin{cases} \geq \ln(\lambda) & \Rightarrow H_1 \\ < \ln(\lambda) & \Rightarrow H_0 \end{cases} \quad (50)$$

and

$$\left(\frac{1}{\sigma_w^2} - \frac{1}{\sigma_w^2 + \sigma_s^2}\right) \sum_{n=0}^{N-1} |x|^2 \begin{cases} \geq \ln(\lambda) - N \cdot \ln\left(\frac{\sigma_w^2}{(\sigma_w^2 + \sigma_s^2)}\right) & \Rightarrow H_1 \\ < \ln(\lambda) - N \cdot \ln\left(\frac{\sigma_w^2}{(\sigma_w^2 + \sigma_s^2)}\right) & \Rightarrow H_0 \end{cases} \quad (51)$$

which leads to the detector $T(x)$ being given as

$$T(x) = \sum_{n=0}^{N-1} |x|^2 \begin{cases} \geq \lambda' & \Rightarrow H_1 \\ < \lambda' & \Rightarrow H_0 \end{cases} \quad (52)$$

where

$$\lambda' = \frac{\ln(\lambda) - N \cdot \ln\left(\frac{\sigma_w^2}{(\sigma_w^2 + \sigma_s^2)}\right)}{\frac{1}{\sigma_w^2} - \frac{1}{\sigma_w^2 + \sigma_s^2}} \quad (53)$$

3.5 Performance of a general NP-detector

As derived in section 3.4, the NP-detector is described as:

$$T(x) = \sum_{n=0}^{N-1} |x[n]|^2 \begin{cases} \geq \lambda' & \Rightarrow H_1 \\ < \lambda' & \Rightarrow H_0 \end{cases} \quad (54)$$

From the model building in section 3.1 we know that $x[n]$ is a complex Gaussian variable:

$$T(x) = \sum_{n=0}^{N-1} x_r^2[n] + x_i^2[n] \quad (55)$$

with

$$x_r, x_i \begin{cases} \stackrel{H_1}{\sim} \mathcal{N}\left(0, \frac{\sigma_w^2 + \sigma_s^2}{2}\right) \\ \stackrel{H_0}{\sim} \mathcal{N}\left(0, \frac{\sigma_w^2}{2}\right) \end{cases} \quad (56)$$

leading to

$$T\left(\frac{2x}{\sigma_w^2 + \sigma_s^2}\right) = \sum_{n=0}^{N-1} \underbrace{\left(\frac{2x_r}{\sigma_w^2 + \sigma_s^2}\right)^2}_{\stackrel{H_1}{\sim} \mathcal{N}(0,1)} + \underbrace{\left(\frac{2x_i}{\sigma_w^2 + \sigma_s^2}\right)^2}_{\stackrel{H_1}{\sim} \mathcal{N}(0,1)} \stackrel{H_1}{\sim} \chi_{2N}^2 \quad (57)$$

and

$$T\left(\frac{2x}{\sigma_w^2}\right) = \sum_{n=0}^{N-1} \underbrace{\left(\frac{2x_r}{\sigma_w^2}\right)^2}_{\stackrel{H_0}{\sim} \mathcal{N}(0,1)} + \underbrace{\left(\frac{2x_i}{\sigma_w^2}\right)^2}_{\stackrel{H_0}{\sim} \mathcal{N}(0,1)} \stackrel{H_0}{\sim} \chi_{2N}^2 \quad (58)$$

This Chi-square distributed test statistics can be described with the following PDFs:

$$p(x|H_1) = \frac{2}{\sigma_w^2 + \sigma_s^2} \cdot \frac{1}{2^N \Gamma(N)} \cdot \left(\frac{2}{\sigma_w^2 + \sigma_s^2} T\right)^{N-1} \cdot \exp\left\{-\frac{2}{\sigma_w^2 + \sigma_s^2} \frac{T}{2}\right\} \quad (59)$$

$$= \frac{1}{(\sigma_w^2 + \sigma_s^2)^N (N-1)!} \cdot T^{N-1} \cdot \exp\left\{-\frac{T}{\sigma_w^2 + \sigma_s^2}\right\} \quad (60)$$

and

$$p(x|H_0) = \frac{2}{\sigma_w^2} \cdot \frac{1}{2^N \Gamma(N)} \cdot \left(\frac{2}{\sigma_w^2} T \right)^{N-1} \cdot \exp\left\{-\frac{2}{\sigma_w^2} \frac{T}{2}\right\} \quad (61)$$

$$= \frac{1}{\sigma_w^{2N} (N-1)!} \cdot T^{N-1} \cdot \exp\left\{-\frac{T}{\sigma_w^2}\right\} \quad (62)$$

In order to obtain the receiver operating characteristics (ROC), a threshold λ' has to be calculated, depending on the false-alarm probability (P_{FA}).

$$P_{\text{FA}} = \int_{\lambda'}^{\infty} p(x|H_0) dT \quad (63)$$

$$= \int_{\lambda'}^{\infty} \frac{1}{\sigma_w^{2N} (N-1)!} \cdot T^{N-1} \cdot \exp\left\{-\frac{T}{\sigma_w^2}\right\} dT \quad (64)$$

Here the Chi-squared cumulative distribution function (CDF) can be used.

$$P_{\text{FA}} = \left[\frac{1}{\Gamma(\frac{2N}{2})} \cdot \gamma\left(\frac{2N}{2}, \frac{2T}{2\sigma_w^2}\right) \right]_{\lambda'}^{\infty} \quad (65)$$

$$= \left[\frac{1}{\Gamma(N)} \cdot \gamma\left(N, \frac{T}{\sigma_w^2}\right) \right]_{\lambda'}^{\infty} \quad (66)$$

$$= \left[\frac{1}{(N-1)!} \cdot \gamma\left(N, \frac{T}{\sigma_w^2}\right) \right]_{\lambda'}^{\infty} \quad (67)$$

$$= \frac{\Gamma(N)}{(N-1)!} - \frac{1}{(N-1)!} \cdot \gamma\left(N, \frac{\lambda'}{\sigma_w^2}\right) \quad (68)$$

$$= 1 - \frac{1}{(N-1)!} \cdot \gamma\left(N, \frac{\lambda'}{\sigma_w^2}\right) \quad (69)$$

$$(P_{\text{FA}} - 1)(N-1)! = \gamma\left(N, \frac{\lambda'}{\sigma_w^2}\right) \quad (70)$$

$$\lambda' = \sigma_w^2 \cdot \gamma^{-1}(N, (P_{\text{FA}} - 1)(N-1)!) \quad (71)$$

The detection probability P_D is calculated in a quite similar way, regarding the CDF of hypothesis H_1 .

$$P_D = \int_{\lambda'}^{\infty} p(x|H_1) dT \quad (72)$$

$$= \int_{\lambda'}^{\infty} \frac{1}{(\sigma_w^2 + \sigma_s^2)^N (N-1)!} \cdot T^{N-1} \cdot \exp\left\{-\frac{T}{\sigma_w^2 + \sigma_s^2}\right\} dT \quad (73)$$

$$= \left[\frac{1}{\Gamma(N)} \cdot \gamma\left(N, \frac{T}{\sigma_w^2 + \sigma_s^2}\right) \right]_{\lambda'}^{\infty} \quad (74)$$

$$= 1 - \frac{1}{(N-1)!} \cdot \gamma\left(N, \frac{\lambda'}{\sigma_w^2 + \sigma_s^2}\right) \quad (75)$$

Including the result of λ' in equation 71, we get

$$P_D = 1 - \frac{1}{(N-1)!} \cdot \gamma\left(N, \frac{\sigma_w \cdot \gamma^{-1}(N, (P_{FA}-1)(N-1)!) }{\sigma_w^2 + \sigma_s^2}\right) \quad (76)$$

as equation describing the receiver operating characteristics (ROC).

3.6 Approximate performance of a general NP detector

Since the test statistics $T(x)$ has the structure

$$T(x) = \sum_{n=0}^{N-1} |x[n]|^2 = T(x) = \sum_{n=0}^{N-1} x_r^2[n] + x_i^2[n] := \sum_{k=1}^N z_k \quad (77)$$

and z_k is a random variable with finite mean and variance (in this case it's a chi-square-distribution with two degrees of freedom), the central limit theorem states that test statistics with **large N** can be approximated by

$$T(x) = \sum_{k=1}^N z_k \approx \tilde{T}(x) \sim \mathcal{N} \left(\sum_{k=1}^N \mu_k, \sum_{k=1}^N \sigma_k^2 \right) \quad (78)$$

where μ_k and σ_k^2 are mean and variance of z_k . Since in our case all z_k are a chi-squared variables with two degrees of freedom and the same mean and variance, this can be simplified to

$$\tilde{T}(x) \sim \mathcal{N} (N \cdot \mathbf{E}\{x_r^2 + x_i^2\}, N \cdot \mathbf{Var}\{x_r^2 + x_i^2\}) \quad (79)$$

As stated in section 3.1 we know x_r and x_i are Gaussian variables. Taking the distributions from equation 56 into account, we get

$$\mathbf{E}\{x_r^2 + x_i^2\} = \mathbf{E}\{x_r^2\} + \mathbf{E}\{x_i^2\} \quad (80)$$

$$= 2 \cdot \begin{cases} \frac{\sigma_w^2 + \sigma_s^2}{2} & \text{for } H_1 \\ \frac{\sigma_w^2}{2} & \text{for } H_0 \end{cases} \quad (81)$$

$$= \begin{cases} \sigma_w^2 + \sigma_s^2 & \text{for } H_1 \\ \sigma_w^2 & \text{for } H_0 \end{cases} \quad (82)$$

and

$$\mathbf{Var}\{x_r^2 + x_i^2\} = \mathbf{Var}\{x_r^2\} + \mathbf{Var}\{x_i^2\} \quad (83)$$

$$= \mathbf{E}\{x_r^4\} - \mathbf{E}\{x_r^2\}^2 + \mathbf{E}\{x_i^4\} - \mathbf{E}\{x_i^2\}^2 \quad (84)$$

$$= \mathbf{E}\{x_r^4\} + \mathbf{E}\{x_i^4\} - \mathbf{E}\{x_r^2\}^2 - \mathbf{E}\{x_i^2\}^2 \quad (85)$$

$$= 2 \cdot \begin{cases} 3 \cdot \left(\frac{\sigma_w^2 + \sigma_s^2}{2}\right)^2 - \left(\frac{\sigma_w^2 + \sigma_s^2}{2}\right)^2 & \text{for } H_1 \\ 3 \cdot \left(\frac{\sigma_w^2}{2}\right)^2 - \left(\frac{\sigma_w^2}{2}\right)^2 & \text{for } H_0 \end{cases} \quad (86)$$

$$= \begin{cases} (\sigma_w^2 + \sigma_s^2)^2 & \text{for } H_1 \\ \sigma_w^4 & \text{for } H_0 \end{cases} \quad (87)$$

The Gaussian approximation of the test statistics is therefore

$$\tilde{T}(x) \begin{cases} \stackrel{H_1}{\sim} \mathcal{N}(N(\sigma_w^2 + \sigma_s^2), N(\sigma_w^2 + \sigma_s^2)^2) \\ \stackrel{H_0}{\sim} \mathcal{N}(N\sigma_w^2, N\sigma_w^4) \end{cases} \quad (88)$$

Using this approximated test statistics, we now can calculate approximated detection and false-alarm probabilities

$$P_D = \int_{\lambda'}^{\infty} p(x|H_1) d\tilde{T} = Q\left(\frac{\lambda' - N(\sigma_w^2 + \sigma_s^2)}{\sqrt{N}(\sigma_w^2 + \sigma_s^2)}\right) \quad (89)$$

$$P_{FA} = \int_{\lambda'}^{\infty} p(x|H_0) d\tilde{T} = Q\left(\frac{\lambda' - N\sigma_w^2}{\sqrt{N}\sigma_w^2}\right) \quad (90)$$

where $Q(x)$ is denoting the Gaussian tail function.

3.7 Complexity of the detector

With the approximation in section 3.6, the required number of samples for a desired detection and false alarm probability can be calculated. Starting at equation 90, we find a threshold

$$P_{\text{FA}} = Q \left(\frac{\lambda' - N\sigma_w^2}{\sqrt{N}\sigma_w^2} \right) \quad (91)$$

$$\sqrt{N}\sigma_w^2 \cdot Q^{-1}(P_{\text{FA}}) = \lambda' - N\sigma_w^2 \quad (92)$$

$$\lambda' = \sqrt{N}\sigma_w^2 \cdot Q^{-1}(P_{\text{FA}}) + N\sigma_w^2 \quad (93)$$

that can be inserted into equation 89, leading to

$$P_D = Q \left(\frac{\sqrt{N}\sigma_w^2 \cdot Q^{-1}(P_{\text{FA}}) + N\sigma_w^2 - N(\sigma_w^2 + \sigma_s^2)}{\sqrt{N}(\sigma_w^2 + \sigma_s^2)} \right) \quad (94)$$

$$= Q \left(\frac{\sqrt{N}\sigma_w^2 \cdot Q^{-1}(P_{\text{FA}}) - N\sigma_s^2}{\sqrt{N}(\sigma_w^2 + \sigma_s^2)} \right) \quad (95)$$

$$\sqrt{N}(\sigma_w^2 + \sigma_s^2) \cdot Q^{-1}(P_D) = \sqrt{N}\sigma_w^2 \cdot Q^{-1}(P_{\text{FA}}) - N\sigma_s^2 \quad (96)$$

$$N\sigma_s^2 = \sqrt{N} \cdot (\sigma_w^2 \cdot Q^{-1}(P_{\text{FA}}) - (\sigma_w^2 + \sigma_s^2) \cdot Q^{-1}(P_D)) \quad (97)$$

$$N^2\sigma_s^4 = N \cdot (\sigma_w^2 \cdot Q^{-1}(P_{\text{FA}}) - (\sigma_w^2 + \sigma_s^2) \cdot Q^{-1}(P_D))^2 \quad (98)$$

$$0 = N \cdot ((\sigma_w^2 \cdot Q^{-1}(P_{\text{FA}}) - (\sigma_w^2 + \sigma_s^2) \cdot Q^{-1}(P_D))^2 - N\sigma_s^4) \quad (99)$$

Since the solution $N = 0$ doesn't provide a useful answer to our question, it follows that:

$$0 = (\sigma_w^2 \cdot Q^{-1}(P_{\text{FA}}) - (\sigma_w^2 + \sigma_s^2) \cdot Q^{-1}(P_D))^2 - N\sigma_s^4 \quad (100)$$

$$N\sigma_s^4 = (\sigma_w^2 \cdot Q^{-1}(P_{\text{FA}}) - (\sigma_w^2 + \sigma_s^2) \cdot Q^{-1}(P_D))^2 \quad (101)$$

$$N = \frac{(\sigma_w^2 \cdot Q^{-1}(P_{\text{FA}}) - (\sigma_w^2 + \sigma_s^2) \cdot Q^{-1}(P_D))^2}{\sigma_s^4} \quad (102)$$

is the number of required samples to attain desired values of P_D and P_{FA} .

3.8 Numerical experiments in PU detection

Based on the results of the previous sections, the detector is now implemented the following way:

$$T(x) = \sum_{n=0}^{N-1} |x[n]|^2 \begin{cases} \geq \lambda' & \Rightarrow H_1 \\ < \lambda' & \Rightarrow H_0 \end{cases} \quad (103)$$

with

$$\lambda' = \sigma_w^2 \cdot \gamma^{-1} (N, (P_{\text{FA}} - 1)(N - 1)!) \quad (104)$$

To test the functionality, the detector is applied on individual 100 sequences of length $N = 256$, noise variance $\sigma_w^2 = 1$ and signal variance $\sigma_s^2 = 5$ to detect the presence or absence of the signal. As false alarm probability, both $P_{\text{FA}} = 0.1$ and $P_{\text{FA}} = 0.01$ are considered.

4 Implementation and results

This chapter describes how the tasks described in section 3 are implemented as well as the computed results are discussed.

4.1 Programming language

All implementations are written in Python, which is one of the most popular interpreted high-level programming languages. With the use of numeric and scientific libraries as *numpy* and *scipy* Python recently has gained popularity in science and education as open-source alternative to the MATLAB environment.

For development, Python version 2.7.16 and the Spyder IDE were used. However, all implementations should also run error-free in newer versions (Tested for version 3.7.3).

4.2 Used libraries

In addition to the standard python language, the following Python libraries are used:

All scripts use *numpy* for numerical calculations.

```
import numpy as np
```

For all plotting, *pyplot* as a part of *matplotlib* is used.

```
from matplotlib import pyplot as plt
```

Statistical distributions such as Gaussian and Chi-square are included in *stats* of *scipy*.

```
from scipy import stats as sta
```

4.3 Frequently used functions

Mean and variance estimators: For estimating the true mean and variance of a sequence x with N samples, the following MVU-estimators are used:

$$\hat{\mu} = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \quad (105)$$

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=0}^{N-1} x^2[n] \quad (106)$$

Numpy provides them as:

```
np.mean(x)
```

and

```
np.var(x)
```

for **x** being a *numpy* vector.

Probability density functions: Each distribution in *scipy.stats* has its function calculating the PDF $p(x)$, for this project

```
sta.norm.pdf(x, loc, scale)
```

and

```
sta.chi2.pdf(x, df, loc, scale)
```

were used. While **x** still is a *numpy* vector, **loc** allows to move the PDF along the **x** axis (as if the true mean was changed) and **scale** adjusts the PDF to a desired standard deviation. For the chi-squared distribution, **df** denotes the degrees of freedom.

Tail/survival functions: In order to calculate the so-called tail-/survival functions of a distribution $p(x)$

$$\int_x^{\infty} p(x')dx' \tag{107}$$

scipy.stats provides the functions

```
sta.norm.sf(x, loc, scale)
```

and

```
sta.chi2.sf(x, df, loc, scale)
```

Here, **x** is normally a single value, which the tail-/survival function should be calculated for. All other parameters are the same as for the PDF-calculation functions. Both functions have inverse functions working in a similar way.

4.4 Results

Figure 5 shows histograms of two completely different discrete frequency spectra and their inverse Fourier transformed complex valued time domain correspondents, split in real and imaginary part. The theoretical shape of a continuous Gaussian distribution is also plotted with an orange line for clarity.

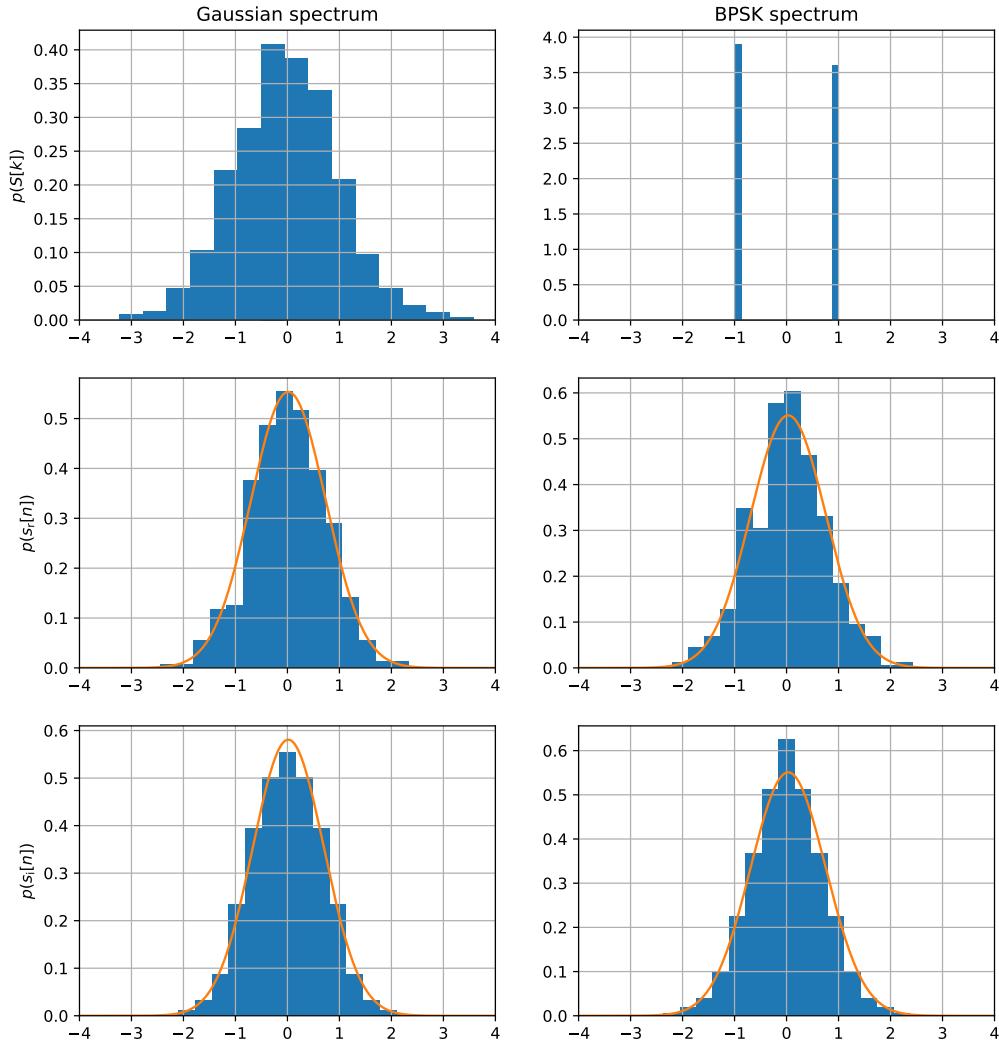


Figure 5: Histograms of the spectrum $S[k]$ and time domain signals $s_r[n]$ and $s_i[n]$ (blue), as well as theoretical Gaussian distributions (orange)

As visible in figure 5, a complex Gaussian approximation of the time domain signal is a quite accurate model, no matter what kind of message encoding is used in the frequency domain. Different estimates of the the time domain signal $s[n]$ is equal to

$E\{s_r[n] \cdot s_i[n]\} = -6.9 \cdot 10^{-18}$ and $E\{s[n]\} = 0.017$ for the Gaussian distribution, and $E\{s_r[n] \cdot s_i[n]\} = 8.5 \cdot 10^{-18}$ and $E\{s[n]\} = 0.031$ for the Binary spectrum.

Figure 6 shows the histogram of the adjusted test statistics for the two test hypotheses H_0 and H_1 , as well as the plot of the true PDF for the chi-squared distribution with 2 degrees of freedom.

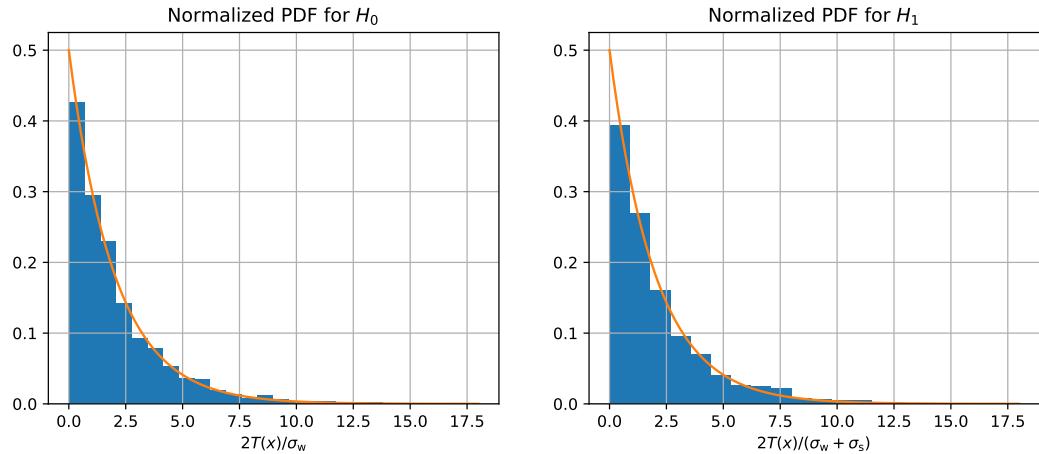


Figure 6: Histograms of the normalised PDF for the two test hypothesis H_0 and H_1 (blue), as well as PDF for the chi-squared distribution with 2 degrees of freedoms (orange).

We can see from figure 6 that the histogram of the adjusted test statistics fits the theoretical values of the chi-squared distribution quite well. The estimates of the variances σ_s^2 and σ_s^2 are equal to $E\{\sigma_w^2\} = 0.995$ and $E\{\sigma_s^2\} = 0.991$

If multiple samples are considered, they lead to more degrees of freedom. Already for $N = 10$ samples, the two PDFs become quite separable compared to the one-sample case. With an increasing number of samples, they also start to resemble Gaussian distributions. This can be seen in figure 7.

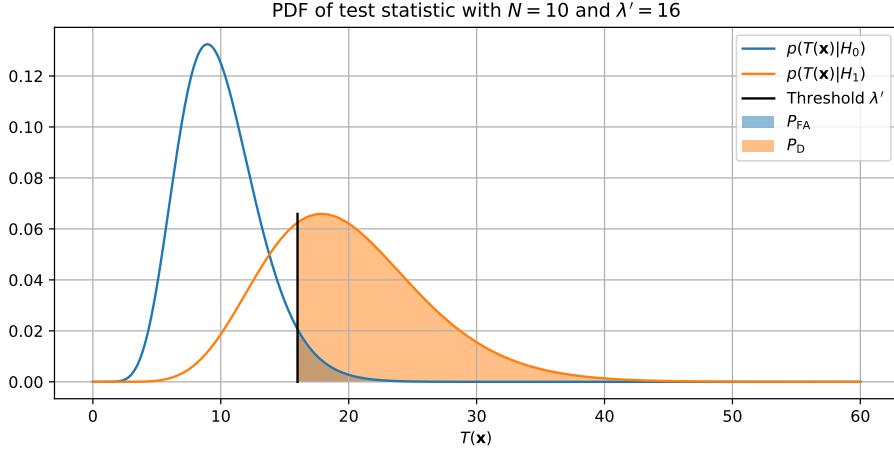


Figure 7: PDF of the test statistics for both hypothesis. The detection and false-alarm probability can be seen as the area under the PDF starting from the threshold.

The effect that the two hypothesis's PDFs become more separable for a growing amount of samples is especially visible in the receiver operating characteristics (ROC), plotted in figure 8. While the test statistics already are quite nice for $N = 10$ samples, they appear almost perfect for $N = 100$ samples.

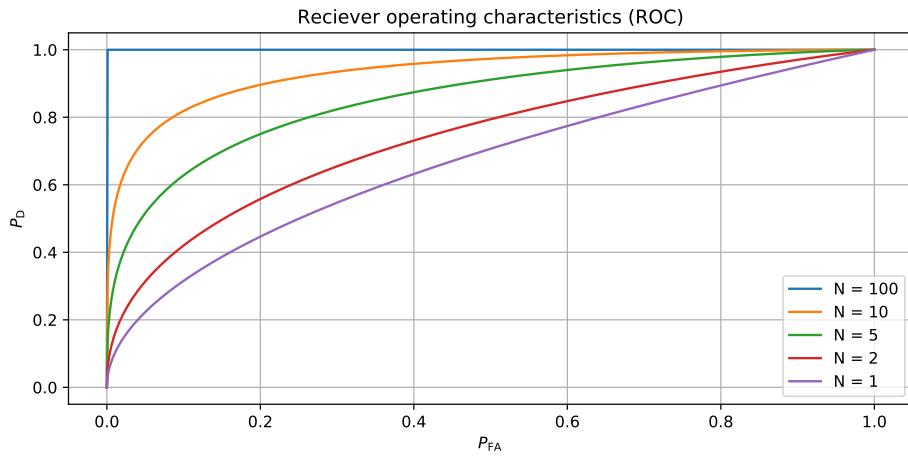


Figure 8: Receiver operating characteristics for test statistics with a different amount of samples

As already stated regarding figure 7, the test statistics look like normally distributed for a big number of samples. The direct comparison of the detection/false alarm probability and the corresponding Gaussian approximations in figure 9 confirm this by showing that the approximation error becomes less and less significant for big numbers of samples. The results can therefore be seen as a practical confirmation of the central limit theorem.

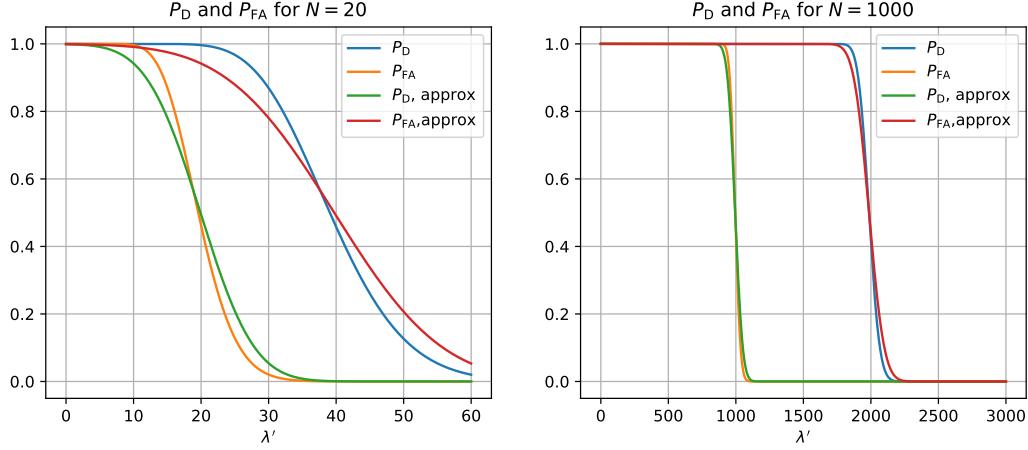


Figure 9: Comparison between the detection/false alarm probabilities and their Gaussian approximations.

Using the Gaussian approximation to calculate the number of required samples as function of the detection probability and the false alarm probability results in the 3d-plot visible in figure 10. While the Gaussian approximation appears to be acceptable for bigger numbers of N (compare with figure 8), ROCs for a low number of samples are even left-curved, standing for counter-productive test statistics. This emphasises clearly that the Gaussian approximation should only be used for a high number of samples.

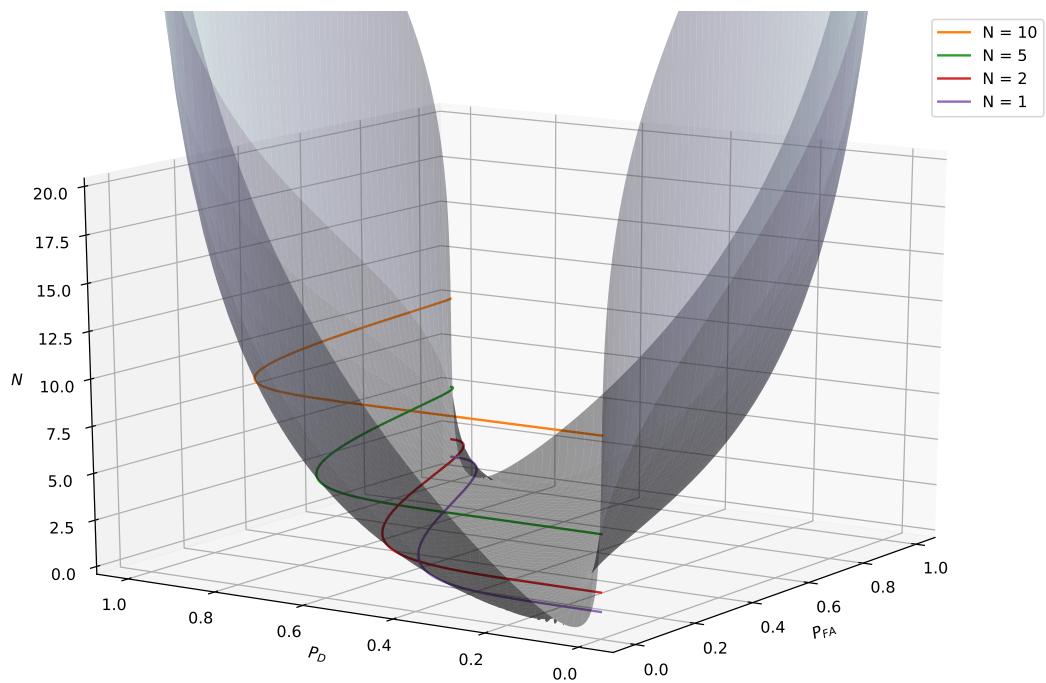


Figure 10: Number of required samples as a function of the detection probability and false alarm probability, computed by using a Gaussian approximation of the test statistics. Height lines can be seen as ROC for the corresponding number of samples.

4.5 Numerical experiment results

Testing the fully implemented detector from section 3.8 with 100 different realisations of $N = 256$ samples each and two false alarm rates, we get the following results:

Realisation	$P_{\text{FA}} = 0.1$	$P_{\text{FA}} = 0.01$	Realisation	$P_{\text{FA}} = 0.1$	$P_{\text{FA}} = 0.01$
0	1	1	30	1	1
1	1	1	31	1	1
2	1	1	32	1	1
3	0	0	33	1	1
4	0	0	34	1	1
5	1	1	35	0	0
6	0	0	36	1	1
7	0	0	37	1	1
8	1	0	38	1	1
9	1	1	39	1	1
10	1	1	40	0	0
11	1	1	41	1	1
12	1	1	42	1	1
13	1	1	43	1	1
14	1	1	44	1	1
15	1	1	45	1	1
16	1	1	46	1	1
17	1	1	47	1	1
18	0	0	48	0	0
19	1	1	49	1	1
20	0	0	50	1	1
21	1	1	51	1	1
22	1	1	52	1	1
23	1	1	53	0	0
24	1	1	54	0	0
25	1	1	55	1	1
26	1	1	56	0	0
27	1	1	57	0	0
28	1	1	58	0	0
29	1	1	59	1	1

Table 1: Testing the detector for different false alarm rates

Realisation	$P_{\text{FA}} = 0.1$	$P_{\text{FA}} = 0.01$	Realisation	$P_{\text{FA}} = 0.1$	$P_{\text{FA}} = 0.01$
60	1	1	80	1	1
61	1	1	81	1	1
62	1	1	82	1	1
63	1	1	83	1	1
64	1	1	84	1	1
65	1	1	85	0	0
66	1	1	86	1	1
67	1	1	87	1	1
68	0	0	88	1	1
69	1	1	89	1	1
70	0	0	90	1	0
71	1	1	91	1	1
72	1	1	92	1	1
73	1	1	93	1	1
74	1	1	94	1	1
75	1	1	95	1	1
76	1	1	96	1	1
77	1	1	97	1	1
78	1	1	98	0	0
79	1	1	99	1	1

Table 2: Testing the detector for different false alarm rates

As visible in the tables 1 and 2, lowering the false alarm rate from 10% to 1%, reduces the number of detections by 2. Assuming that the detector only has 1 out of 100 false alarms in case $P_{\text{FA}} = 0.01$, this would be $\frac{2}{21} = 9.52\%$ less actual false alarms, resembling the detector's theoretical basis quite well.

The corresponding missing probabilities however are extremely low for such a high number of samples per realisation. In fact, Python returns a detection probability of $P_D = 1$ in both cases. Even if a far smaller value for P_{FA} is chosen, the missed detection probability will practically be $P_M = 0$.

Assuming as many samples per realisation and the same variances as for this test set, one can with great confidence assure that a 5G secondary user device will not interfere with a primary device while he still might be able to detect unused bandwidth for most cases.

5 Conclusion

In the work described and discussed in this report the main goal was to derive and implement a Neyman-Pearson Detector to detect, whether a specific bandwidth is used at a time or not. Similar detectors are used for so-called *cognitive radio*, a strategy which allows a slightly more effective use of underutilised resources in cellular networks and therefore is applied in the newest 5G-networking standard.

For this purpose, theoretical basics on statistics and detection theory were explained, a stochastic model for a transmitted signal in noise was developed and a Neyman-Pearson-Detector to detect this signal was derived. Facing non-Gaussian models for the derived test statistics, a Gaussian approximation for these test statistics was made and compared to the non-Gaussian chi-squared model.

To verify all derivations and assumptions, the models were implemented in Python and tested with given measurement data. As a result, real measurement data tests resembled the derived models quite well, the chi-squared distribution appears to be the right model to describe the used test statistics. A Gaussian approximation of these test statistics works properly as well, but just for a high number of test samples. As long the signal noise ratio is sufficient, the assembled detector is performing stunningly well, especially with a big amount of samples.

Conclusively, a Neyman-Pearson-Detector of the kind described in the report is a useful tool for implementing a *cognitive radio* system in reality and might be a small but important part of a faster, cheaper and more reliable next-generation networking standard.

Developing this project gave us an interesting insight on applying theoretical matter on real problems, working in groups, managing our time and resources and structure our work in professional ways.

References

- [1] Pierluigi Salvo Rossi *TTT4275 – Estimation, Detection and classification Project description - Spectrum Sensing in OFDM Cognitive Radios*, 2020, NTNU
- [2] Stefan Werner *TTT4120 – Digital Signal Processing 2018 Discrete Random Signals*, 2018, NTNU
- [3] Magne H. Johnsen *TTT4275 – Estimation, Detection and Classification Lecture: Introduction to detection*, 2020, NTNU
- [4] Tor A. Myrvoll, Stefan Werner and Magne H. Johnsen *TTT4275 – Estimation, Detection and classification theory*, 2020, NTNU
- [5] Arak M. Mathai and Hans J. Haubold *Probability and Statistics. Walter de Gruyter GmbH*, 2010, ISBN 978-3-11-056253-8
- [6] Edouard Geffray and Jean-Bernard Auby *The political and legal consequences of smart cities*, 2017 from <https://journals.openedition.org/factsreports/4281> visited on 04-30-2020,12.23