Detection Project Report

Group 61
Student Kenny Hoang Nguyen 478374
Student Martin Lund Haug xxxxx

April, 2020

NTNU – Trondheim
Norwegian University of
Science and Technology

Abstract

A short summary using about half a page about:

i The course/project

ii the results

iii conclusion

Contents

1 Introduction

Detection problems occurs in many places in everyday life, in your computer you have electrical signals which the CPU has to detect as 1 or 0, or in other words a current is present or not. Another example is in air defense where the military has radars to detect if there is an hostile air action present or not. In this project we are going to look at a cognitive radio system with primary users (PU) and secondary users (SU), where the SUs are going to detect if there are PUs in the network so that they can decide if they can utilize the frequency spectrum without interfering with the quality of service (QoS) of the PUs.

This problem where the SUs use the frequency spectrum in an opportunistic manner whenever there are IDLE PUs is interesting since in wireless communication systems, spectrums are a scarce resource that service providers pay a substantial amount of money to the government in order to license the spectrum. This cost is covered by the customers in their monthly mobile subscription cost. Paying customers demands a certain QoS, which in the mentioned case is the PUs, and any interference appearing on the communication channel should be kept at a minimum to deliver the promised QoS.

This project starts by introducing the theoretical background necessary to understand and solve this problem in section 2. Then the tasks that needs to be solved for this problem is in section 3 followed by the implementation and results in section 4 then finally it is all wrapped up in 5.

Here we should write about

i The goals/motivation of the course/project.

ii Why is your task of general interest to society? etc. . .

iii How the report is organized

- In chapter x the theory is described
- in chapter y the implementation is described
- . . .
- and finally the conclusion is given in chapter z

You cite by using [2]

2 Theory

2.1 Gaussian Distribution

The gaussian distribution or the normal distribution is an important distribution that is often used in natural and social sciences for real-valued random variables when their distributions are not known. The importance of this distribution comes from the central limit theorem, that states for any under some conditions the average of many (enough) observations of a random variable with finite mean and variance converges to a normal distribution even if the random variable comes from another distribution.[4]. The most important thing here is that the probability density function of a normal distribution with mean μ and variance σ^2 is

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{1}$$

which gives the probability to obtain any value $x \in \mathbb{R}$ from this distribution. In this particular project we use the complex gaussian distribution which for our random variables has the probability density function

$$f(x) = \frac{1}{\pi\sigma^2} e^{-\frac{1}{\sigma^2}|x-\mu|^2} \tag{2}$$

Which in this case take in any value $x \in \mathbb{C}$

2.2 Estimators

When we with probable cause can say something about the distribution that the random variables are sampled from, but not their mean and/or variance, we can estimate these distributions properties. Assumed that the random variables are sampled independent from the same identical distribution (iid), then the mean can be estimated with the average of the samples, which is unbiased.

$$\begin{aligned}
\hat{\mu} &= \mathbb{E}\{\frac{1}{N}\sum_{n=0}^{N-1} x\} \\
&= \frac{1}{N}\sum_{n=0}^{N-1} \mathbb{E}\{x\} \\
&= \frac{1}{N}\sum_{n=0}^{N-1} \mu \\
&= \mu \qquad (3)
\end{aligned}$$

Meaning that the expected value of the mean of the samples will, with the number of samples taken, converge to the actual expected value of the distribution.

This is also the case if there are samples of the variance of the data available

$$\begin{aligned}\hat{\sigma^2} &= \mathbb{E}\{\frac{1}{N}\sum_{n=0}^{N-1}\sigma^2\} \\ &= \frac{1}{N}N\sigma^2 \\ &= \sigma^2 \qquad (4)\end{aligned}$$

These two results are used when solving the problems later on in this report.

2.3 (Binary) Hypothesis Testing

Detection problems are often formulated to be about if a signal is present or not in conditions which masks the signal that we desire to detect, for instance white gaussian noise. The hypothesises that could be formulated is:

$$\begin{aligned}\text{Null hypothesis} H_0 : x[n] \sim \mathbb{P}_0 \\ \text{Alternative hypothesis} H_1 : x[n] \sim \mathbb{P}_1\end{aligned} \qquad (5)$$

where the null hypothesis is ”no signal present” and alternative hypothesis is ”signal present”, and $\mathbb{P}_0$ and $\mathbb{P}_1$ are two arbitrary distributions.
Let $[x[0]x[1]\dots x[N-1]]$ be sampled random variables from a sensor where it is constant "1" when detecting an object and constant "0" when not detecting anything. A simple detector in this case could be by setting a threshold at $\lambda = 0.3$ that detects when the threshold is broken. The problem with this simple detector appears when the signal from our sensor is noisy. Let now the samples from our sensor be

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

where $w[n]$ $\mathcal{N}(0,1)$. Then under the null hypothesis in a non-noisy environment $w[n] = 0 \implies x[n] = 0$, however we have a noisy envirnoment so $x[n]$ obtains random values from the normal distribution with zero mean and unit variance. In figure 1 we can see how the data we obtain from 100 samples may look like. From these 100 samples there are 39 samples that are above the set threshold meaning that from these 100 samples we will get 39 false positives/alarms which will alert the user that there is an object detected when there really is not. Our goal when solving detection problems is to develop a detector that from the samples can find a decision rule so that the number of false alarms is minimized but at the same time manages to detect correctly when there is an object present.[3]

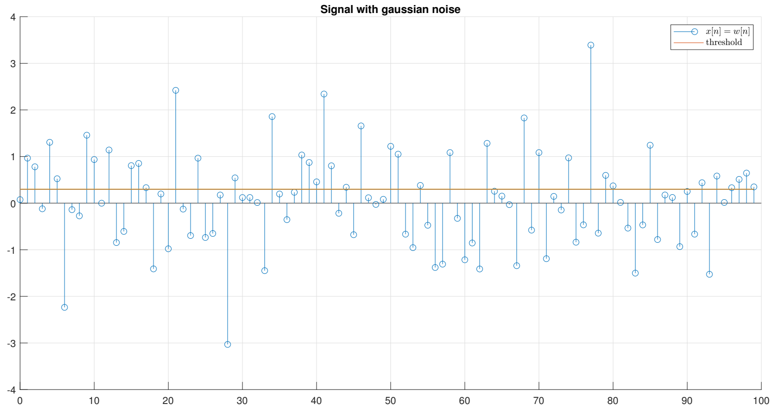


Figure 1: 100 samples from a gaussian distribution $\mathcal{N}(0,1)$

2.4 Neyman-Pearson detector

The Neyman-Pearson detector utilizes the likelihood ratio test (LRT)

$$L(\mathbf{x}) \triangleq \frac{p_1(\mathbf{x})}{p_0(\mathbf{x})} \left\{ \geq \lambda \implies H_1 < \lambda \implies H_0 \right. \tag{6}$$

where $p_1(x)$ and $p_0(x)$ is the point distribution functions or the likelihood functions of the distributions under H_1 and H_0 respectively. With the Neyman-Pearson detector the threshold λ is chosen to satisfy the constraint and to maximize the power such that

$$P_{FA} = \alpha = \int_{L(\mathbf{x})>\lambda} p_0(\mathbf{x}) d\mathbf{x} \tag{7}$$

where α_0 the tuning factor. This goes hand in hand with the detection rate of the detector as well since the Neyman-pearson lemma states that to increase the power of the likelihood ratio test, the false alarm rate is also increased.

Mye dårlige formuleringer her. Må skrives om

i Inform the reader that this chapter is a presentation of the theory needed to understand the task

ii You may copy parts from lecture notes (but inform the reader that you have done this!). Also refer to books in former courses or other literature

- Always refer to the source and
- Use quotation marks when quoting (copying)

iii Use figures if possible/natural!

3 The tasks

The main problem that is going to be solved in this project is the detection problem

$$\begin{aligned} &H_0 : x(n) = w(n), n = 0, 1, \dots, N-1 \\ &H_1 : x(n) = s(n) + w(n), n = 0, 1, \dots, N-1 \end{aligned} \tag{8}$$

where $s(n)$ is a waveform sequence of the PU and $w(n)$ is an additive white complex gaussian noise. To construct the sequence $s(n)$, that is transmitted over the wireless channel, the modulation method orthogonal frequency-division multiplexing (OFDM) is used. Each information symbol $S(k)$, $k = 0, 1, \dots, N-1$ is allocated on each N carrier frequencies. The unique time-domain signal $s(n)$ corresponding to the sampled spectrum is obtained by inverse discrete fourier transform. Thus is the PUs time-domain signal given by:

$$s(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S(k) e^{\frac{j2\pi nk}{N}}, n = 0, 1, \dots, N-1 \tag{9}$$

which we notice is a complex-valued quantity.

3.1 Task 1: Model building

Here the data $x[n]$ is generated, which can be used in our analysis to obtain a suitable detector. To generate $x[n]$ we need do to verify that the complex-valued time-domain OFDM signal sequence $s(n) = s_R(n) + js_I(n)$ is independent and identically distributed, in addition to verify that it is accurately modelled with a complex gaussian distribution.
Here the datasets T1 are used as the information symbol sent over. The two datasets contains samples that are taken from a standard normal Gaussian distribution and binary phase shift keying respectively.

3.2 Task 2: One-sample detector

In this task only a single sample is used to generate the NP-detector. That is there needs to be done calculations to retrieve the decision rule that decides for when to choose null hypothesis over the alternative hypothesis and vice versa.

3.3 Task 3: Performance of the one-sample detector

Here the datasets T3 are given and are to be used to verify that

$$H_0 : \frac{2x_R^2(0)}{\sigma_w^2} + \frac{2x_I^2(0)}{\sigma_w^2} = \frac{2|x(0)|^2}{\sigma_w^2} \tag{10}$$

$$H_1 : \frac{2x_R^2(0)}{\sigma_w^2 + \sigma_s^2} + \frac{2x_I^2(0)}{\sigma_w^2 + \sigma_s^2} = \frac{2|x(0)|^2}{\sigma_w^2 + \sigma_s^2} \tag{11}$$

are χ^2-distributed with 2 degrees of freedom. Using this we can approximate that $\frac{2|x(0)|^2}{\sigma^2}$ has the point-distribution function $f(x) = \frac{1}{2}e^{-\frac{x}{2}}$ for $x > 0$. Which is more manageable than the pdf for the complex gaussian so that it can be used to easily find the probability of detection and false alarm.

3.4 Task 4: NP detector with data set of K samples

Here the one-sample detector derived in the previous tasks are expanded so it can be used when there are $K > 1$ samples. Then the threshold λ' is derived so that in this case probability of detection is maximized and probability of false alarm is less than a given limit α.

ubsectionTask 5: Performance of the general NP detector First compute the distribution of the test statistic obtained in the previous task under hypothesis H_0 and H_1. Plot the receiving operating characteristics.

3.5 Task 6: Approximate performance of the general NP detector

Central limit theorem to approximate the characteristic test statistic as a gaussian random variable and compute the PDF of the test statistic. Plot P_D and P_{FA} as a function of the threshold.

3.6 Task 7: Complexity of the detector

Use approximation in task 6 and find an expression to compute the number of samples required to attain a given P_{FA} and P_D.

3.7 Task 8: Numerical experiments in PU detection

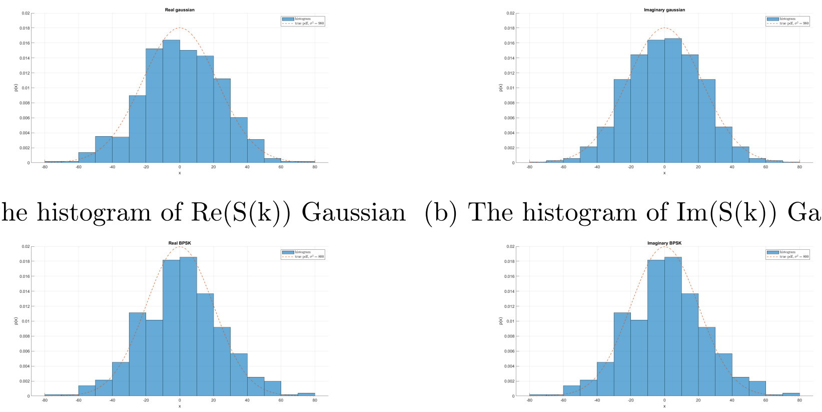
Given a dataset of numerical experiment data, apply the NP detector to decide wether a PU is present or not. Discuss the results

i Give a short description of all tasks and why are we doing these tasks

ii Describe the task

4 Implementation and results

4.1 Task 1: Model development



(a) The histogram of Re(S(k)) Gaussian (b) The histogram of Im(S(k)) Gaussian

(c) The histogram of Re(S(k)) BPSK (d) The histogram of Im(S(k)) BPSK

Figure 2: The histograms compared to pdf of gaussian

4.2 Task 2: One-sample-detector

In this task only one sample is used meaning that the detection problem is in this case:

$$\begin{aligned} H_0 &: x(0) = w(0) \\ H_1 &: x(0) = s(0) + w(0) \end{aligned}$$

It is given from the problem that $w \sim \mathbb{C}\mathcal{N}(0, \sigma_w^2)$ and $s \sim \mathbb{C}\mathcal{N}(\mu_s, \sigma_s^2)$ which means that under H_0, x is from same distribution as w and thus a complex gaussian with same mean and variance as w. Under H_1, which is a sum of two complex gaussian distribution, x also is from a complex gaussian distribution, however the mean and variance needs to be calculated. The mean of $x(0)$ under H_1 is

Usikker på om det er vits med disse utregningene

$$\begin{aligned} \mathbb{E}\{x(0)\} &= \mathbb{E}\{s(0) + w(0)\} \\ &= \mathbb{E}\{s(0)\} + \mathbb{E}\{w(0)\} \\ &= \mu_s + 0 = \mu_s \end{aligned}$$

The variance of $x(0)$ under H_1 is

$$\begin{aligned} \mathrm{Var}\{x(0)\} &= \mathrm{Var}\{s(0) + w(0)\} \\ &= \mathrm{Var}\{s(0)\} + \mathrm{Var}\{w(0)\} \\ &= \sigma_s^2 + \sigma_w^2 \end{aligned}$$

Meaning that

$$p(x; H_0) = p_0(x) = \frac{1}{\sigma_w^2 \pi} e^{-\frac{|x|^2}{\sigma_w^2}} \tag{12}$$

$$p(x; H_1) = p_1(x) = \frac{1}{(\sigma_w^2 + \sigma_s^2)\pi} e^{-\frac{|x-\mu_s|^2}{\sigma_s^2+\sigma_w^2}} \tag{13}$$

Setting up the LRT:

$$\begin{aligned} L(x) = \frac{p_1(x(0))}{p_0(x(0))} &= \frac{\frac{1}{(\sigma_w^2+\sigma_s^2)\pi} e^{-\frac{|x(0)-\mu_s|^2}{\sigma_s^2+\sigma_w^2}}}{\frac{1}{\sigma_w^2 \pi} e^{-\frac{|x(0)|^2}{\sigma_w^2}}} \\ &= \frac{\sigma_w^2}{\sigma_w^2 + \sigma_s^2} e^{-\frac{1}{\sigma_w^2+\sigma_s^2}|x(0)-\mu_s|^2 + \frac{1}{\sigma_w^2}|x(0)|^2} \end{aligned}$$

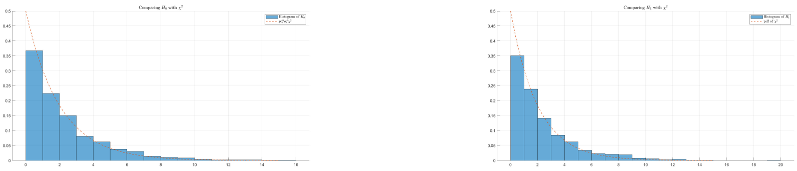
Since it was shown in task 1 that $\mu_s \simeq 0$ this can be used to simplify the caluclations. The decision rule is to choose H_1 when $L(x) \geq \lambda$, since $L(x(0))$ is a monotonically increasing function then the inequality holds for $\ln L(x(0)) \geq \ln \lambda$.

$$\begin{aligned} \ln L(x(0)) = \ln(\sigma_w^2) - \ln(\sigma_w^2 + \sigma_s^2) - \frac{1}{\sigma_w^2 + \sigma_s^2}|x(0)|^2 + \frac{1}{\sigma_w^2}|x(0)|^2 &\geq \ln \lambda \\ (\frac{1}{\sigma_w^2} + \frac{1}{\sigma_s^2 + \sigma_w^2})|x(0)|^2 &\geq \ln \lambda - \ln(\frac{\sigma_w^2}{\sigma_w^2 + \sigma_s^2}) \\ |x(0)|^2 = x_R(0)^2 + x_I(0)^2 \geq \frac{\sigma_w^2(\sigma_w^2 + \sigma_s^2)}{\sigma_s^2}(\ln \lambda - \ln(\frac{\sigma_w^2}{\sigma_w^2 + \sigma_s^2})) &= \lambda' \end{aligned}$$

This also means that the decision rule is to choose H_1 when $|x(0)|^2 \geq \lambda'$ and choose H_0 when $|x(0)|^2 < \lambda'$ where

$$\lambda' = \frac{\sigma_w^2(\sigma_w^2 + \sigma_s^2)}{\sigma_s^2}(\ln \lambda - \ln(\frac{\sigma_w^2}{\sigma_w^2 + \sigma_s^2})) \tag{14}$$

4.3 Task 3: Performance of the one-sample detector



(a) The histogram of (10)

(b) The histogram of (11)

The probability for false alarm is the probability that $|x(0)|^2 > \lambda'$, assumed that it is actually the null hypothesis that is correct. Instead of using the complex gaussian pdf we have shown that $2\frac{|x(0)|^2}{\sigma^2}$ under H_0 and H_1 can be approximated to be χ^2 distributed meaning that

$$\begin{aligned}
P_{FA} &= \text{Prob}\{|x(0)| \geq \lambda' | H_0\} \\
&= \frac{\sigma_w^2}{2} \int_{\lambda'}^{\infty} \frac{e^{-\frac{x}{2}}}{2} dx \\
&= -\frac{\sigma_w^2}{2} e^{-\frac{x}{2}} \Big|_{\lambda'}^{\infty} \\
&= \frac{\sigma_w^2}{2} e^{-\frac{\lambda'}{2}}
\end{aligned}$$

Which only holds when $\lambda' > 0$.
Furthermore the probability for detection is

$$\begin{aligned}
P_D &= \text{Prob}\{|x(0)| \geq \lambda' | H_1\} \\
&= \frac{\sigma_w^2 + \sigma_s^2}{2} \int_{\lambda'}^{\infty} \frac{e^{-\frac{x}{2}}}{2} dx \\
&= -\frac{\sigma_w^2 + \sigma_s^2}{2} e^{-\frac{x}{2}} \Big|_{\lambda'}^{\infty} \\
&= \frac{\sigma_w^2 + \sigma_s^2}{2} e^{-\frac{\lambda'}{2}}
\end{aligned}$$

Which also only holds for $\lambda' > 0$

4.4 Task 4: Generalized for multiple samples

We take the one-sample-detector a step further and generalize it to use it for multiple samples. That is that the likelihood function is used to calculate the LRT so that we get

$$\begin{aligned}
\ln L(x) &= \ln \prod_{n=0}^{K-1} \frac{1}{(\sigma_w^2 + \sigma_s^2)\pi} e^{-\frac{|x(n)|^2}{\sigma_s^2 + \sigma_w^2}} - \ln \prod_{n=0}^{K-1} \frac{1}{\sigma_w^2 \pi} e^{-\frac{|x(n)|^2}{\sigma_w^2}} \\
&= \sum_{n=0}^{K-1} \left(\ln \frac{\sigma_w^2}{\sigma_w^2 + \sigma_s^2} + \frac{|x(n)|^2}{\sigma_w^2} - \frac{|x(n)|^2}{\sigma_w^2 + \sigma_s^2} \right) \\
&= \sum_{n=0}^{K-1} \left(\ln \frac{\sigma_w^2}{\sigma_w^2 + \sigma_s^2} + \frac{\sigma_s^2}{\sigma_w^2(\sigma_w^2 + \sigma_s^2)} |x(n)|^2 \right) \geq \ln \lambda
\end{aligned}$$

So the decision rule ends up being to choose H_1 when

$$\sum_{n=0}^{K-1} |x(n)|^2 \geq \frac{\sigma_w^2(\sigma_w^2 + \sigma_s^2)}{\sigma_s^2} \left(\ln \lambda - K \ln \frac{\sigma_w^2}{\sigma_w^2 + \sigma_s^2} \right) = \lambda' \qquad (15)$$

i What is this chapter about

ii Matlab implementation

iii Any specific Matlab m-commands used?

iv A flow-diagram is recommended

v Results (use figures/tables if possible)

vi Discussion of results

vii Matlab code (documented) in appendix

5 Conclusion

i An extended version of summary

- theory, programming, implementation, and so on.

ii Both positive and negative results should be mentioned

iii Include main points from discussion of results

iv Any positive/negative comments on the task intention and quality...

v What you learnt from the project

Table 1: Parameters and values.

Symbol	Parameter	Value	Unit
l_c	Distance from elevation axis to counterweight	0.50	m
l_h	Distance from elevation axis to helicopter head	0.64	m
l_p	Distance from pitch axis to motor	0.18	m
K_f	Force constant motor	0.25	N/V
J_e	Moment of inertia for elevation	0.83	kg m^2
J_λ	Moment of inertia for travel	0.83	kg m^2
J_p	Moment of inertia for pitch	0.034	kg m^2
m_h	Mass of helicopter	1.05	kg
m_p	Motor mass	1.81	kg
m_c	Counterweight mass	0.73	kg

6 General LaTeX tips

Some tips were given in Section 1, and this section will elaborate with some more concrete examples. Also check out the source files for some additional useful packages.

6.1 Matrix Equations

Here is a matrix equation you can use as a template:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & -b & 0 & 0 & 0 \\ -a & 1 & 0 & 0 & 0 & -b & 0 & 0 \\ 0 & -a & 1 & 0 & 0 & 0 & -b & 0 \\ 0 & 0 & -a & 1 & 0 & 0 & 0 & -b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} ax_0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{16}$$

6.2 Tables

If you want, you can use the source for Table 1 to see how a (floating) table is made. Variables and symbols are always in italics, while units are not. Generating large, complicated tables can get very tedious. Luckily there exists some tools that can assist the table generation, see e.g. `http://www.tablesgenerator.com/`.

6.3 The `\input{}` command

By using `\input{whatever}` in your main tex file (`labreport.tex` in this case), the content of `whatever.tex` will be included in your pdf. This way you can split the contents into different files, e.g. one for each problem of the assignment. This makes it easier to restructure the document, and arguably improves the readability of the tex files. For instance; maybe you want each problem to start on a new page? Simply add \newpage before each `\input{}` command. Alternatively, you can use the `\include{}` command to achieve more or less the same effect. See [5] for more information.

6.4 Citations and Reference Management

In academic writing, it is very important to cite your sources. In Latex this is done by defining an an entry in a *BibTeX* bibliography file like this (from `bibliography.bib`):

```
@book{Chen2014,
  title={Linear System Theory and Design},
  author={Chen, Chi-Tsong},
  isbn={9780199964543},
  year={2014},
  publisher={Oxford University Press, Incorporated}
}
```

and then using the `\cite` command in your Latex document. For instance `\cite{Chen2014}` will produce [2].

There are many different citation styles, and a lot of customization that is possible, so please check out e.g. [1, 6][1].

There is also a lot of useful software to manage your references. Some popular examples include JabRef (`http://www.jabref.org/`), Mendeley (`https://www.mendeley.com/`) and EndNote. JabRef is perhaps the simplest of these three, and stores all information in a `.bib` file that you can directly use in your Latex document. Both Mendeley and EndNote can export references as BibTeX.

6.5 listings

The `listings` package makes it easy to include code in the report. For example listing 1 includes code that is written in the tex file. You can also specify what the code listings should look like: color, line numbers, frames...

This is great! However, try to keep the amount of code in the report to a reasonable level, and remember; code in itself is not an explanation.

[1] Keep citation of web pages to a minimum, and consider using `http://web.archive.org` if you are worried that the reference may change or be removed in the future.

Listing 1: Some Matlab code, with the source in the tex file

```
degree = 6;
out = ones(size(X1(:,1)));
for i = 1:degree
   for j = 0:i
  out(:, end+1) = (X1.^(i-j)).*(X2.^j);
   end
end
```

6.6 todonotes

The `todonotes` package is great for work in progress. Few things are more embarrassing than forgetting to remove “Remember to fix this before delivery!!!!!!” from the middle of your report. Instead, use `\todo{Remember to fix this before delivery!!!!!!}` . This will show up like a red box in the margin. Some prefer `\todo[inline]{FIXME2!!!}` which produces

FIXME2!!!

Remember to fix this before delivery!!!!!!

You can also use \listoftodos to get a list of all the todos in your document, and \missingfigure will create a dummy figure, like fig. 4, that you can replace once you have made a proper figure. This way you can start referencing figures/plots before you make them, and still be reminded that you need to make them.

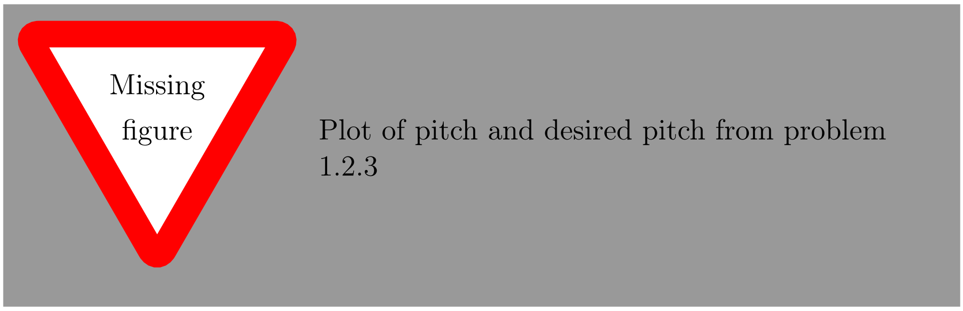


Figure 4: Pitch and desired pitch

When you are finished with your report (or have run out of time) you can simply change \usepackage{todonotes} to \usepackage[disable]{todonotes} and they will all magically disappear!

6.7 cleveref

The observant reader might have noticed the use of `\cref` in referencing tables, figures etc. This is a bit more clever than the normal `\ref` because it detects what you are referencing based on the prefix of the label. Then it prints the appropriate “prefix”. So `\cref{fig:my_awesome_fig}` will produce fig. 4, whereas `\cref{tab:parameters}` will produce table 1. Notice how the labels of the table and the figureare prefixed with `tab:` and `fig:` respectively. If you want it to say e.g. “figure” instead of “fig.”, this is completely customizable. There is also `\Cref` for a capitalized version.

References

[1] *bibtex vs. biber and biblatex vs. natbib.* `http://tex.stackexchange.com/questions/25701/bibtex-vs-biber-and-biblatex-vs-natbib`. Accessed: 2017-02-13.

[2] Chi-Tsong Chen. *Linear System Theory and Design.* Oxford University Press, Incorporated, 2014.

[3] Tor A. Myrvoll, Stefan Werner, and Magne H. Johnsen. *Estimation, Detection and Classification Theory.* 2020.

[4] *Normal distribution.* `https://en.wikipedia.org/wiki/Normal_distribution`. Accessed: 2020-04-19.

[5] *When should I use* `input` *vs.* `include`*?* `http://tex.stackexchange.com/questions/246/when-should-i-use-input-vs-include`. Accessed: 2017-02-13.

[6] *Wikibooks LaTeX.* `https://en.wikibooks.org/wiki/LaTeX`. Accessed: 2016-08-30.

A MATLAB code

A.1 Problem 1

```
clc; close all;


%% Confirm that our data is iid and normal gaussian
S_k_gauss         = load('Dataset/T1_data_Sk_Gaussian.mat').T1_data_Sk_
S_k_bpsk          = load('Dataset/T1_data_Sk_BPSK.mat').T1_data_Sk_BPSK

[N, one]        = size(S_k_gauss);

%{
s_gauss         = zeros(N, 1);
s_bpsk          = zeros(N, 1);

%% Generate the signal s(n) using DFT
for n = 0:N-1
    for k = 0:N-1
        % Remember that matlab is 1-indexed
        sum_S_gauss = S_k_gauss(k+1)*exp(1i*2*pi*n*k/N);
        sum_S_bpsk  = S_k_bpsk(k+1)*exp(1i*2*pi*n*k/N);
    end

    %s(n) in time domain as given in project description
    s_gauss(n+1)    = sum_S_gauss/sqrt(N);
    s_bpsk(n+1)     = sum_S_bpsk/sqrt(N);
end
%}
s_gauss             = fft(S_k_gauss);
s_bpsk              = fft(S_k_bpsk);

s_gauss_real        = real(s_gauss);
s_gauss_imag        = imag(s_gauss);

s_bpsk_real         = real(s_bpsk);
s_bpsk_imag         = imag(s_bpsk);


%% Estimate the expected values
sr_expected_gauss           = sum(s_gauss_real)/N;
si_expected_gauss           = sum(s_gauss_imag)/N;
s_expected_gauss            = sr_expected_gauss + 1i*si_expected_gauss;
```

```matlab


sr_expected_bpsk            = sum(s_bpsk_real)/N;
si_expected_bpsk            = sum(s_bpsk_imag)/N;
s_expected_bpsk             = sr_expected_bpsk + 1i*sr_expected_bpsk;

% When samples are taken uniformly from the same population with any
% distribution the expected value of the mean is always unbiased
product_s_expected_gauss    = sum(s_gauss_real.*s_gauss_imag)/N;
product_s_expected_bpsk     = sum(s_bpsk_real.*s_bpsk_imag)/N;

disp(['The expected value of s_n with gaussian samples: ' ...
    num2str(s_expected_gauss)]);


disp(['The expected value of s_n with bpsk samples: ' ...
    num2str(s_expected_bpsk)]);

disp(['The expected value of product of real and imaginary part' ...
    ' with gaussian samples: ' num2str(product_s_expected_gauss)]);

disp(['The expected value of product of real and imaginary part' ...
    ' with bpsk samples: ' num2str(product_s_expected_bpsk)]);

%% Gaussian Create a generic complex point distribution
sigma_s_sq      = 980; %10^(-5);

px_r            = makedist('Normal', 'mu', 0, 'sigma', sqrt(sigma_s_sq/
px_i            = makedist('Normal', 'mu', 0, 'sigma', sqrt(sigma_s_sq/

x = (-80:0.1:80)';%(-6*10^(-3):10^(-4):6*10^(-3))';


%% Plot histograms of the sampled data Gaussian
figure(1);
title('Real gaussian');
hold on
histogram(s_gauss_real, 'Normalization', 'pdf');
hold on
plot(x, px_r.pdf(x), '--', 'Linewidth', 1)
hold on
xlabel('x');
hold on
ylabel('p(x)');
```

```
hold on
grid on;
legend('histogram', 'true pdf, $\sigma^2 = 980$', 'Interpreter', 'latex
hold off;

figure(2);
title('Imaginary gaussian');
hold on
histogram(s_gauss_imag, 'Normalization', 'pdf');
hold on
plot(x, px_i.pdf(x), '--', 'Linewidth', 1)
hold on
xlabel('x');
hold on
ylabel('p(x)');
hold on
grid on;
legend('histogram', 'true pdf, $\sigma^2 = 980$', 'Interpreter', 'latex
hold off;

%% BPSK Create a generic complex point distribution

sigma_s_sq      = 800; %5*10^(-4);

px_r            = makedist('Normal', 'mu', 0, 'sigma', sqrt(sigma_s_sq/
px_i            = makedist('Normal', 'mu', 0, 'sigma', sqrt(sigma_s_sq/

x = (-80:0.1:80)'; %(-0.05:0.001:0.05)';
%px              = px_r.pdf(x).*px_i.pdf(x);

figure(3);
title('Real BPSK');
hold on
histogram(s_bpsk_real, 'Normalization', 'pdf');
hold on
plot(x, px_r.pdf(x), '--', 'Linewidth', 1)
hold on
xlabel('x');
hold on
ylabel('p(x)');
hold on
grid on;
legend('histogram', 'true pdf, $\sigma^2 = 800$', 'Interpreter', 'latex
hold off;
```

```matlab

figure(4);
title('Imaginary BPSK');
hold on
histogram(s_bpsk_real, 'Normalization', 'pdf');
hold on
plot(x, px_i.pdf(x), '--', 'Linewidth', 1)
hold on
xlabel('x');
hold on
ylabel('p(x)');
hold on
grid on;
legend('histogram', 'true pdf, $\sigma^2 = 800$', 'Interpreter', 'latex
hold off;
```

A.2 Problem 3

```matlab
close all; clc


%% Load data
x_h0         = load('Dataset/T3_data_x_H0.mat').T3_data_x_H0; % x=w
x_h1         = load('Dataset/T3_data_x_H1.mat').T3_data_x_H1; % x=w+s

sigma_w_sq   = load('Dataset/T3_data_sigma_w.mat').w;
sigma_s_sq   = load('Dataset/T3_data_sigma_s.mat').s_t;

[N, ones]    = size(x_h0);

%% Estimate the variances sigma_w_sq and sigma_s_sq
sigma_w_sq_hat   = sum(abs(sigma_w_sq))/N;
sigma_s_sq_hat   = sum(abs(sigma_s_sq))/N;


%% Calculate the histogram for our data
chi_sq_h0    = zeros(N, 1);
chi_sq_h1    = zeros(N, 1);

for i = 1:N
    chi_sq_h0(i) = 2*abs(x_h0(i))^2/sigma_w_sq_hat;
    chi_sq_h1(i) = 2*abs(x_h1(i))^2/(sigma_s_sq_hat+sigma_w_sq_hat);
end

```

```


%% Generate an arbitrary chi-square random variable with two DoF

x            = 0:0.1:15;
doF          = 2;
chi_sq       = pdf('Chisquare', x, doF)';


figure(1);
title('Comparing $H_0$ with $\chi^2$', 'Interpreter', 'latex');
hold on
histogram(chi_sq_h0, 'Normalization', 'pdf');
hold on
plot(x, chi_sq, '--', 'Linewidth', 1);
hold on
grid on;
hold on;
legend('Histogram of $H_0$','$pdf of \chi^2$', 'Interpreter', 'latex');
hold off

figure(2);
title('Comparing $H_1$ with $\chi^2$', 'Interpreter', 'latex');
hold on
histogram(chi_sq_h1, 'Normalization', 'pdf');
hold on
plot(x, chi_sq, '--', 'Linewidth', 1);
hold on
grid on;
hold on;
legend('Histogram of $H_1$', 'pdf of $\chi^2$', 'Interpreter', 'latex')
hold off
```