

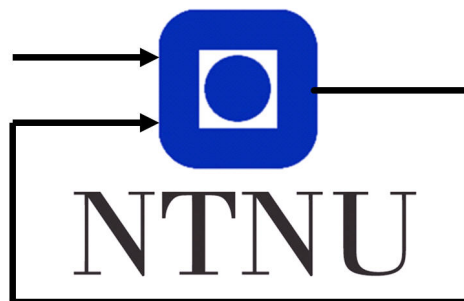
LaTeX Lab Report Template

Group 61

Student Kenny Hoang Nguyen 478374

Student Martin Lund Haug xxxxxx

April, 2020



Department of Engineering Cybernetics

Abstract

A short summary using about half a page about:

- i The course/project
- ii the results
- iii conclusion

Contents

1	Introduction	1
2	Theory	2
3	The task	3
4	Implementation and results	4
5	Conclusion	5
6	General LaTeX tips	6
6.1	Matrix Equations	6
6.2	Tables	6
6.3	The <code>\input{}</code> command	6
6.4	Citations and Reference Management	7
6.5	listings	7
6.6	todonotes	7
6.7	cleveref	8
	References	10
	Appendix	11
A	MATLAB Code	11
A.1	plot_constraint.m	11
B	Simulink Diagrams	11
B.1	A Simulink Diagram	11

1 Introduction

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

- 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 task

- i Give a short description of all tasks and why are we doing these tasks
- ii Describe the task

4 Implementation and results

- 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:

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

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 [3] 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 entry in a *BibTeX* bibliography file like this (from `bibliography.bib`):

```

1 @book{Chen2014,
2   title={Linear System Theory and Design},
3   author={Chen, Chi-Tsong},
4   isbn={9780199964543},
5   year={2014},
6   publisher={Oxford University Press, Incorporated}
7 }
```

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, 4]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.

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 deliv-

¹Keep 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

```

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

```

ery!!!!!!” 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
deliv-
ery!!!!!!

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. 1, 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.

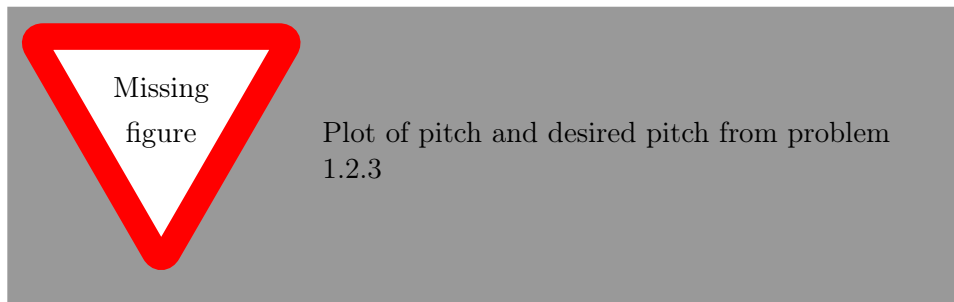


Figure 1: Pitch and desired pitch

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. 1, whereas `\cref{tab:parameters}` will produce table 1. Notice how the labels of the table and the figure are 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] *When should I use `\input` vs. `\include`?* <http://tex.stackexchange.com/questions/246/when-should-i-use-input-vs-include>. Accessed: 2017-02-13.
- [4] *Wikibooks LaTeX*. <https://en.wikibooks.org/wiki/LaTeX>. Accessed: 2016-08-30.

A MATLAB Code

This section should contain your MATLAB code. DO NOT attach files posted online (that you didn't write). Note that the method used to input code below does not look as pretty when the lines are too long.

A.1 plot_constraint.m

```
1 % Plot a figure with some Latex in the labels
2 l = linspace(70,170)*pi/180;
3 a = 0.2;
4 b = 20;
5 l_b = 2*pi/3;
6
7 e = a*exp(-b*(l-l_b).^2);
8
9 l_deg = l*180/pi;
10 e_deg = e*180/pi;
11
12 figure(1)
13 plot(l_deg,e_deg, 'LineWidth', 2)
14
15 handles(1) = xlabel('$\lambda$/degrees');
16 handles(2) = ylabel('$e$/degrees');
17 set(handles, 'Interpreter', 'Latex');
```

B Simulink Diagrams

This section should contain your Simulink diagrams. Just like the plots, these should be in vector format, like in Figure 2. Make them tidy enough to understand.

B.1 A Simulink Diagram

Figure 2 shows a Simulink diagram. You can use the `print_simulink.m` function, included in the source code repository for this document, to export a Simulink model to EPS.

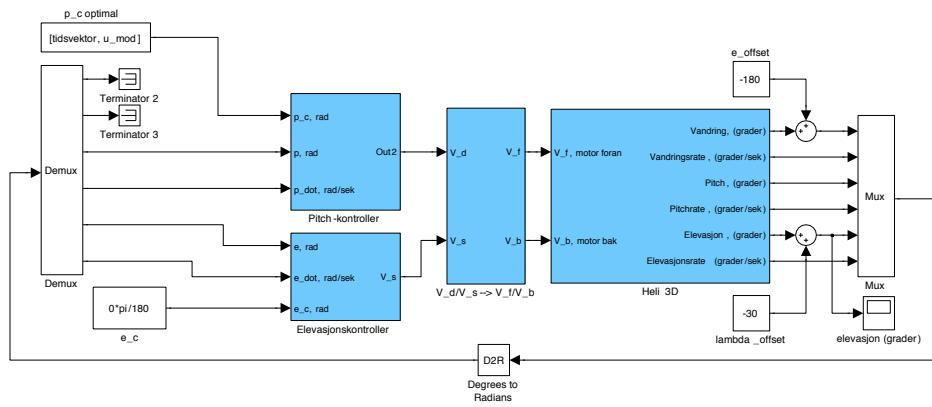


Figure 2: A Simulink diagram.