LaTeX Lab Report Template

Group 00
Student 70000
Student 70001
Student 70002

January 1, 1970

Department of Engineering Cybernetics

Abstract

This document outlines a few important aspects of a lab report. It contains some advice on both content and layout. The LaTeX source for this document is also published, and you can use it as a template of sorts for your own report. You can find an up to date version of the source at `https://github.com/ntnu-itk/labreport`. The main file, "labreport.tex", defines the structure of the document. The "preamble.tex" file is the document preamble, and contains a lot of informative comments. The document is based on work done by Tor Aksel Heirung for TTK4135, and is now under continuous improvement by Andreas L. Flåten and Kristoffer Gryte (happily accepting suggestions and contributions from the community).

When you write your own report, this section (the abstract) should contain a *very* short summary of what the lab is about and what you have done.

Contents

1 Introduction

Your introduction should contain an overview of the work you were assigned, as well as a few sentences putting the work into a larger perspective. You should also give a quick description of how the report is organized (as is done below).

You should of course put most of the work into doing good work in the lab and then presenting it in the report. When presenting your work in the report, both content and presentation/layout matters. Since your only way of communicating your good effort in the lab is through writing about it here, the way you write about it is essential. This means that even if you have the very best controller but describe it poorly, you will probably not be rewarded for the good results. A plot showing perfect control is worth very little if it is not accompanied by a clear description of what it represents.

Layout is naturally less important than content, but it still matters. You can think of report writing like selling an apartment; when you present your apartment for potential buyers you will of course clean the apartment and make it good looking. How clean the apartment is does of course not determine its value, but it is still important since it influences the subjective value your buyers will put on the apartment.

1.1 Software

You are of course free to use whatever software you want for report writing. You can also submit a handwritten report, although this is probably not a great idea if your handwriting can be hard to read.

You can also use Word or a similar word processor. However, it is next to impossible to achieve decent layout with Word. The support for vector graphics (discussed later) is extremely poor, and text tends to look pretty bad (bad support for kerning and ligatures). Furthermore, math is both time consuming and difficult to input, and tends to look very ugly. In general, a report written in Word looks like a draft.

It is strongly recommended to use Latex. Unless you tweak the layout too much, your report will almost certainly look very good. Although it can take a bit of effort to get started, it is also much quicker to use than Word and similar programs. The support for math and vector graphics is also great.

If you are new to Latex, you can have a look at the source for this document to get started. You can also look at the presentation by [1] (in Norwegian) or consult [5]. Another good reason to learn Latex is that you probably don't want to write your master's thesis in something like Word, doing so would likely be very frustrating. Being reasonably fluent in Latex before you get that far will make your thesis work much smoother.

Some of you are probably fluent in Latex and might plan to write the report using it. Please resist the temptation (if any) to change the fonts, make super fancy headers (they are not necessary for a report like this), change the margins, change the paragraph indentation and/or spacing, and similar things.

A great tool for collaborating on Latex documents is ShareLaTeX at `https://www.sharelatex.com/`; if you use this you won't have to install anything on your computer. Texmaker at `http://www.xm1math.net/texmaker/` is a good cross-platform editor. Some people like Lyx, which is a Latex editor that behaves a little bit like Word. If you prefer to compile your Latex document on the command line, the latexmk `https://www.ctan.org/pkg/latexmk` command is a great tool included in most TeX distributions. There is also a simple Vim plugin that uses latexmk as its backend called LaTeX-BoX `https://github.com/LaTeX-Box-Team/LaTeX-Box`.

1.2 Other Comments

Unless you have a very good reason not to, you should write the report in English. If you have problems with Latex, the solution is usually just a few Google searches away.

This report is organized as follows: Section 2 contains some course specific equations, and some tips on how to create illustrations. Several LaTeX tips can be found in Section 3, such as how to create a table and matrix equations. Section 4 contains some advice on using plots from MATLAB. The closing remarks are in Section 5, respectively. Appendix A contains a MATLAB file while Appendix B shows an example Simulink diagram. The Bibliography can be found at the end, on page 16.

2 Problem Description

You should have a section that describes the lab setup, including a model of the helicopter. If you want, you can copy the source code for the model equations:

$$J_p\ddot{p} = L_1 V_d \tag{1}$$

$$J_e\ddot{e} = L_2\cos(e) + L_3 V_s\cos(p) \tag{2}$$

$$J_\lambda\ddot{\lambda} = L_4 V_s\cos(e)\sin(p) \tag{3}$$

Since these equations belong together, it's a good idea to number them like this:

$$J_p\ddot{p} = L_1 V_d \tag{4a}$$

$$J_e\ddot{e} = L_2\cos(e) + L_3 V_s\cos(p) \tag{4b}$$

$$J_\lambda\ddot{\lambda} = L_4 V_s\cos(e)\sin(p) \tag{4c}$$

You can then both reference individual equations ("the elevation equation Equation (4b)") or reference the entire model ("the linear model Equation (4)"). Regardless of your choice of software, never hard-code a reference, always use dynamic references.

You could also align the equations like this:

$$\begin{aligned} J_p\ddot{p} &= L_1 V_d \qquad (5a)\\ J_e\ddot{e} &= L_2\cos(e) + L_3 V_s\cos(p) \qquad (5b)\\ J_\lambda\ddot{\lambda} &= L_4 V_s\cos(e)\sin(p) \qquad (5c) \end{aligned}$$

You can consult [4] for more about writing math.

2.1 Illustrations

If you decide to include an illustration, that's great. You can in general copy figures and illustrations from the textbook, the assignement text, or other places. However: ALWAYS CITE THE SOURCE. You can also draw your own (cite the source if it is heavily based on someone else's.). Figures 1 and 2 was created quickly with Ipe (`http://ipe.otfried.org/`). Inkscape is a good alternative for more advanced illustrations. Some people prefer the Latex package TikZ (`http://texample.net/tikz/examples/`), but this takes a little effort to learn.

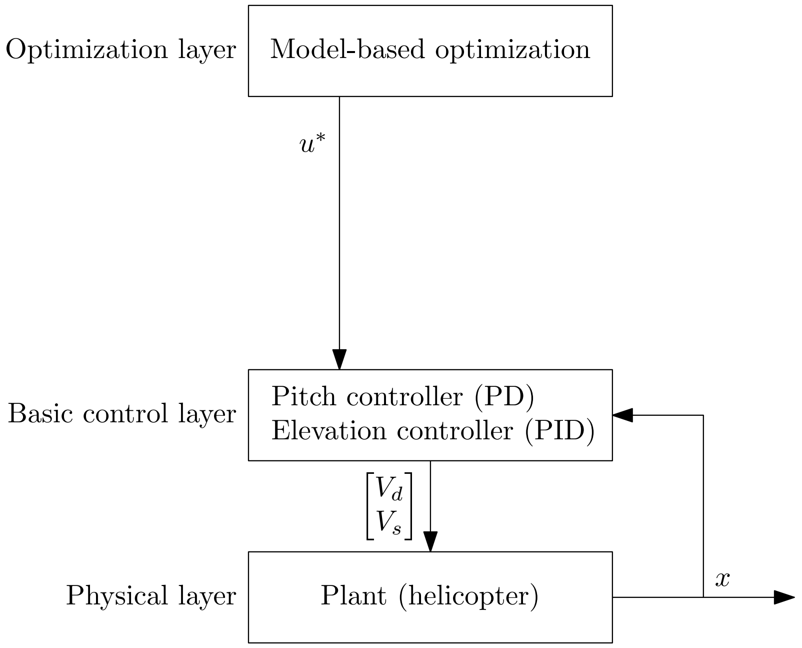

Figure 1: A figure created with Ipe for TTK4135.

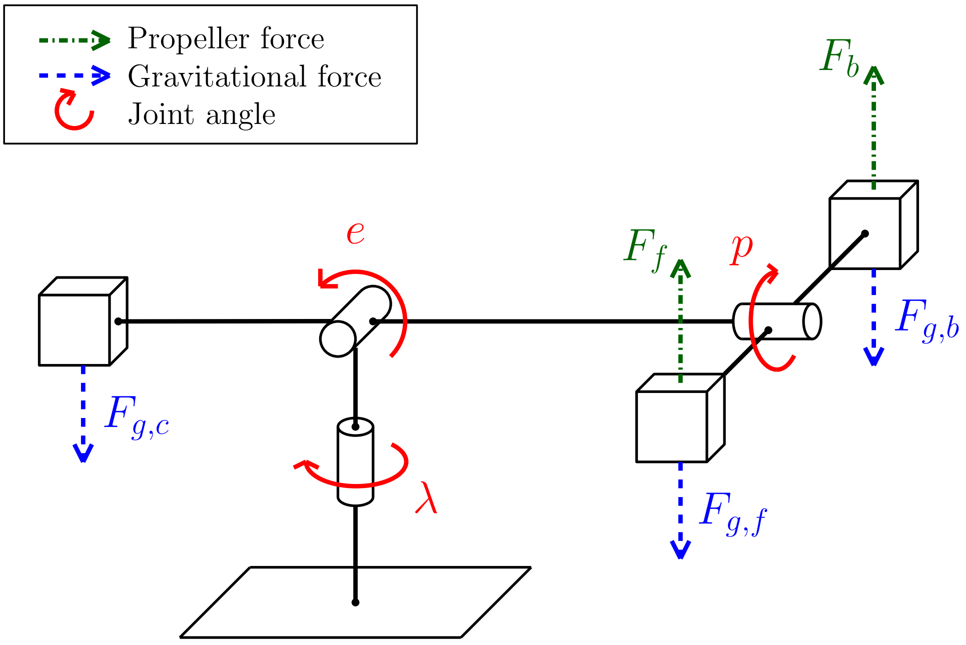

Figure 2: A figure created with Ipe for TTK4115.

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

3 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.

3.1 Matrix Equations

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

$$\begin{bmatrix} 1 & 0 & 0 & 0 & -b & 0 & 0 & 0 \\ -a & 1 & 0 & 0 & 0 & -b & 0 & 0 \\ 0 & -a & 1 & 0 & 0 & 0 & -b & 0 \\ 0 & 0 & -a & 1 & 0 & 0 & 0 & -b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} ax_0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6}$$

3.2 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/`.

3.3 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 [6] for more information.

3.4 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 [3].

There are many different citation styles, and a lot of customization that is possible, so please check out e.g. [2, 7][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.

3.5 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.

3.6 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-

[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
```

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 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. 3, 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.

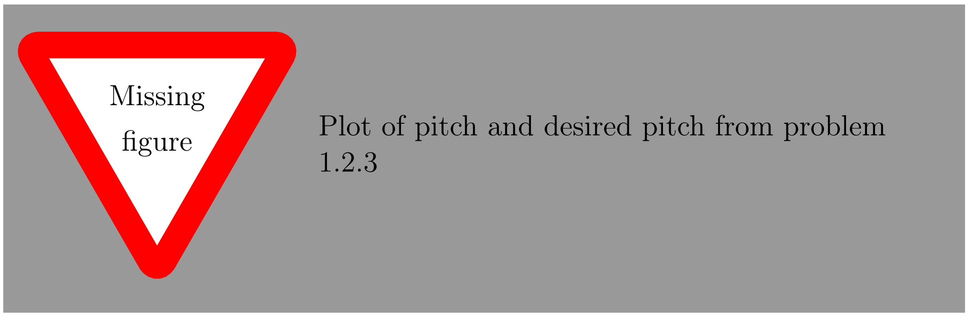

Figure 3: Pitch and desired pitch

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!

3.7 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. 3, 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.

4 Results and Figures

Answer all the parts of the exercise in an organized and clear manner. You should of course try to get good results in all the exercises, but if you have made a good effort without achieving great performance, a good discussion of possible reasons is just as good. Present your thinking and efforts and discuss possible reasons for good or bad results.

Include plots and/or tables of all relevant results, but make sure you don't overwhelm the reader with too many plots. Have a clear plan about what you want to communicate with a specific plot/figure, and use appropriate labels and comments. Keep in mind that the plots should be as "readable" as possible; that is, they should not be too hard to interpret and be reasonably self contained.

There are some important things to consider when exporting figures from MATLAB, most importantly which format you use. Never ever use JPEG for anything that is not a photography or similar. Any figure, like a plot or block diagram, must never be stored as a JPEG. If you zoom in on Figure 4 you can see a lot of noise close to any of the dark curves and lines, this is due to the compression in JPEG. Figure 4 will look horrible both on a screen and on paper.

The PNG format is slightly better for plots, but since it is a raster format (a grid of pixels), it looks ugly if you zoom in. It also looks ugly if you scale it, both on a screen and on paper. Try to avoid PNG if you can. Figures 5 and 6 are both PNG figures; the latter being a larger figure scaled more than the former. Note both how choppy and ugly the blue curve is, and how the different sizes create inconsistent font sizes.

The simplest way to get a reasonably good looking plot is to save it as EPS in MATLAB. Do this by clicking "File" in the figure window, and the "Save As..."; choose "EPS file (*.eps)" in the "Save as type:" menu.[2] Figure 7 shows a plot in EPS format. Since EPS is a vector format, the Figure can be scaled and still look good (but mind the font size!). If you zoom in you can see that the curve and the letters/numbers are smooth. A figure in vector format will usually look good both on a screen and on paper.

Note that the size of the actual figure window in MATLAB determines how large the exported figure is. Hence, if you enlarge the figure window before exporting, you will need to scale the figure by a larger factor in the report. This will lead to a tiny font in the figure. There are many better ways of exporting graphics from MATLAB, but they quickly become fairly involved. The above method of exporting to EPS will in most cases give nice figures.

You can write Latex in your MATLAB figures. The script used to create

[2] pdfLatex does not support EPS directly, but since we have loaded the *epstodf* package, this is not a problem.

Figures 4 and 7 is included in Appendix A.1. Do not use a screen shot of a scope of figure in MATLAB in your report.

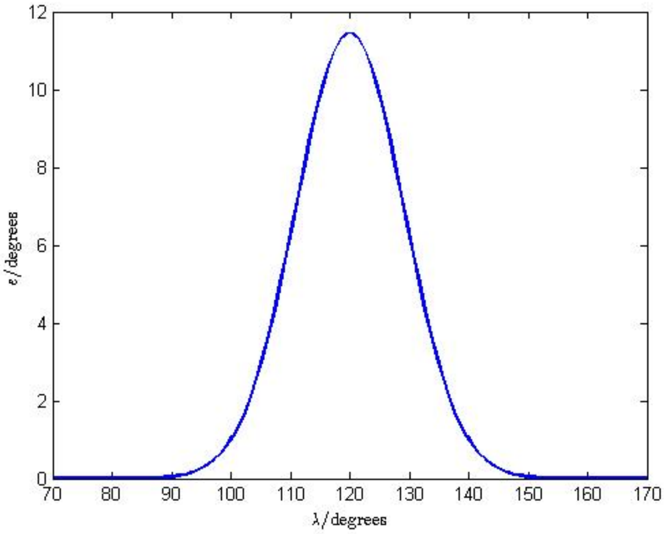

Figure 4: A plot in JPEG format — a very bad idea.

Remember to reference all figures in the text. Figures have a number and should be referenced by that number (again, always use dynamic references). They also tend to float around, meaning they generally don't appear where you ask them to in the text. This is fine, do not try to force a figure (or a table) to appear in a particular place. As long a you refer to it, it's easy to find. No figure should be included without being referenced in the text.

If you look at the source code for including figures, you can see that the optional option `[htb]` has been used. This tells Latex where you wish the figure to appear, in prioritized order. `h` means "Here", t means "Top of this page", b means "Bottom of this page", and p (not used here) means "on a Page with only floats (such as figures and tables)". Note that your wish might not be granted, and this is because Latex actually optimizes the placement of figures. If you start forcing figures to be in specific places, it often leads to really strange layout somewhere else in the document.

Generally, let Latex handle the documentation layout. This is one of the main reasons to choose Latex over software such as Microsoft Word.

4.1 Results and Discussion

All problems should have their own discussion of results.

Remember: all plots and results need a description, explanation, and discussion.

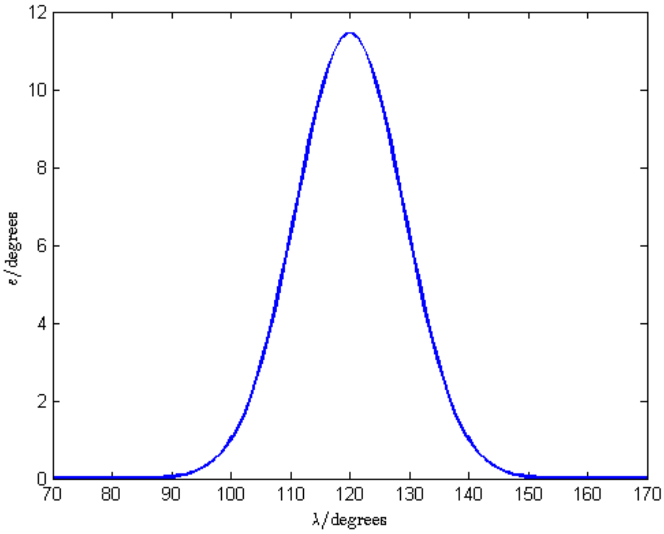

Figure 5: A plot in PNG format — a bad idea.

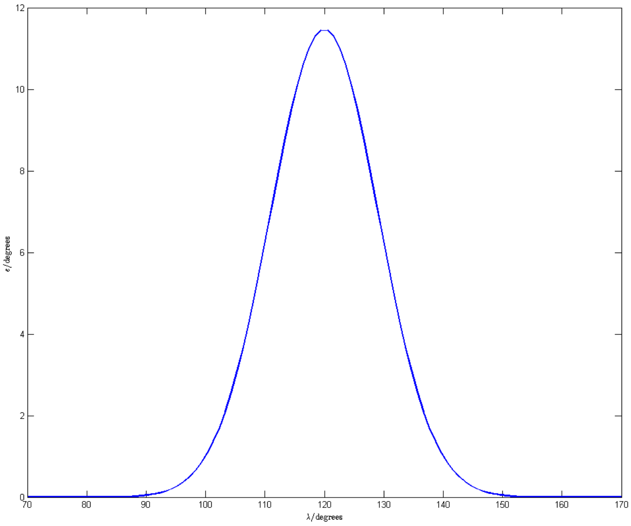

Figure 6: A plot in PNG format — a bad idea. This figure is originally larger than the other PNG figure, but both are scaled to the same size.

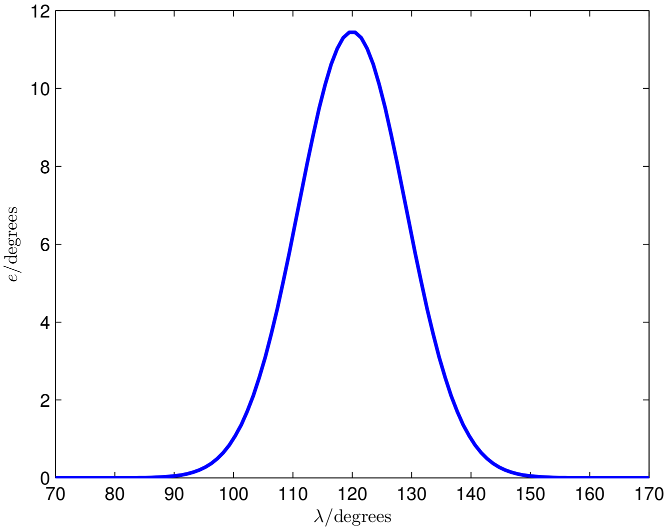

Figure 7: A plot in EPS format — a much better idea.

5 Conclusion

This does not have to be long, but try to write a few reasonable closing remarks.

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

```
% Plot a figure with some Latex in the labels
l = linspace(70,170)*pi/180;
a = 0.2;
b = 20;
l_b = 2*pi/3;

e = a*exp(-b*(l-l_b).^2);

l_deg = l*180/pi;
e_deg = e*180/pi;

figure(1)
plot(l_deg,e_deg, 'LineWidth', 2)

handles(1) = xlabel('$\lambda$/degrees');
handles(2) = ylabel('$e$/degrees');
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 8. Make them tidy enough to understand.

B.1 A Simulink Diagram

Figure 8 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.

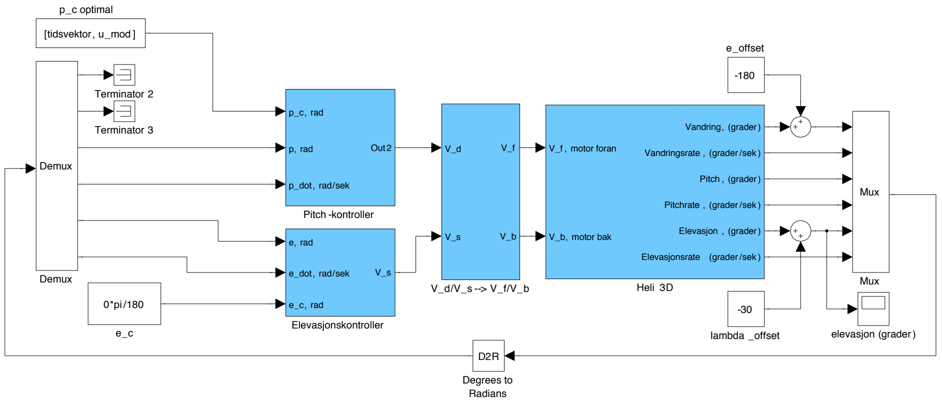

Figure 8: A Simulink diagram.

References

[1] H Berland. *En introduksjon til Latex*. `http://www.pvv.ntnu.no/~berland/latex/latexintro4up.pdf`. 2010.

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

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

[4] M. Downes. *Short Math Guide for LATEX*. `ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf`. 2002.

[5] T. Oetiker et al. *The Not So Short Introduction to LATEX 2e*. `http://tobi.oetiker.ch/lshort/lshort.pdf`. 2011.

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

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