

# Your Long Thesis Title Goes Here and May Extend Over Two Lines

Your Name Here BE (Hons 1)

A thesis submitted in fulfillment  
of the requirements of the degree of  
Doctor of Philosophy



THE UNIVERSITY OF  
SYDNEY

Australian Centre for Field Robotics  
School of Aerospace, Mechanical and Mechatronic Engineering  
The University of Sydney

Submitted March 2013; revised July 2013

# Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

**Your Name Here**

22 July 2013

# Abstract

Abstract text goes here...

If you've just opened up this template, you should check out [Chapter 1](#) for a quick introduction. Note also that hyperlinks are rendered in colour for convenience during editing—see `../LaTeX/packages.tex`. Note also that there is a simple way to change all of the coloured hyperlinks to black in `../LaTeX/packages.tex`—look for `\usepackage[colorlinks]`.

This paragraph is coloured in “highlightcolour”, defined in `../LaTeX/commands.tex`. To get rid of all highlighting in the document you can just redefine `highlightcolour`—see `\definecolorhighlightcolor` in `../LaTeX/commands.tex`. Also have a look at `\definecolor{todocolour}` and `\definecolor{TODOcolour}` in the same file.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Acknowledgements

This thesis would not have been possible without the support of a great many of my work colleagues, family and friends . . .

You could dedicate the thesis to someone, or write something poignant. . .

*Hydrogen, in sufficient quantity and given enough time,  
will end up writing, reviewing and hopefully citing this thesis.*

# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>ix</b>
<b>List of Algorithms</b>	<b>x</b>
<b>List of Theorems</b>	<b>xi</b>
<b>Nomenclature</b>	<b>xii</b>
<b>Glossary</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Structure of the Template . . . . .	2
1.2 Personalising the Thesis . . . . .	5
1.3 Chapters and Appendices . . . . .	5
1.3.1 Renaming Chapters . . . . .	5
1.3.2 Re-ordering Chapters . . . . .	7
1.4 Single- or Double-Sided Page Printing? . . . . .	8
1.5 Organisation of the Thesis Template . . . . .	8

---

<b>2</b>	<b>Literature Review</b>	<b>10</b>
2.1	A Bit on Citations . . . . .	10
2.2	Cross-Referencing Within Your Document . . . . .	13
<b>3</b>	<b>Example Chapter on ‘Perception’ Stuff</b>	<b>15</b>
3.1	Figures . . . . .	15
3.1.1	Figure Placement . . . . .	18
3.2	Tables . . . . .	19
3.3	Equations . . . . .	20
3.3.1	Learning to Set Equations . . . . .	22
3.4	Acronyms . . . . .	22
3.5	Summary . . . . .	23
<b>4</b>	<b>Some Estimation Stuff</b>	<b>24</b>
4.1	Theorems and Hypotheses . . . . .	24
4.1.1	Solution Consistency . . . . .	26
4.2	Examples . . . . .	27
4.3	Algorithms . . . . .	28
4.4	Summary . . . . .	28
<b>5</b>	<b>Experiments That Prove Things</b>	<b>30</b>
5.1	Example Section . . . . .	30
5.2	Example Section Two . . . . .	30
5.3	Third Example Section . . . . .	30
5.4	Example Section Four . . . . .	31
5.5	Summary . . . . .	31
<b>6</b>	<b>Conclusion</b>	<b>33</b>
6.1	Summary . . . . .	33
6.2	Contributions . . . . .	33
6.3	Future Work . . . . .	34

---

<b>List of References</b>	<b>35</b>
<b>A An Example Appendix</b>	<b>37</b>
A.1 Code Listings . . . . .	37
A.2 Multi-Page Tables . . . . .	38
A.3 Landscape Tables . . . . .	39
A.4 Including the PDF of a Relevant Paper . . . . .	40



# List of Figures

3.1	Example figure. . . . .	16
3.2	Vehicle Coordinate Frames . . . . .	17
3.3	Circular Polar Measurement . . . . .	17
4.1	Example Side-by-Side Figures . . . . .	28

# List of Tables

3.1	Example Table . . . . .	19
3.2	A More Complex Table Example . . . . .	20
A.1	Page-Spanning ‘Super Table’ . . . . .	38
A.2	Table in Landscape Orientation . . . . .	44

# List of Algorithms

4.1 Cost Cell Expansion . . . . .	29
-----------------------------------	----

# List of Theorems

4.1 Hypothesis (Descartes) . . . . .	25
4.1 Theorem (Zeroth Law of Thermodynamics) . . . . .	25
4.2 Theorem (First Law of Thermodynamics) . . . . .	25
4.3 Theorem (Second Law of Thermodynamics) . . . . .	25
4.4 Theorem (Third Law of Thermodynamics) . . . . .	25
4.1 Definition (Marginal) . . . . .	26
4.1 Proposition (Solution Consistency) . . . . .	26
4.2 Proposition (Solution Mapping) . . . . .	26
4.1 Example (A Basic Demonstration) . . . . .	27

# Nomenclature

## List of Symbols

v Variable Name, units

## List of Acronyms

**ACFR** Australian Centre for Field Robotics

**INS** inertial navigation system

**GPS** global positioning system

**DGPS** differentially-corrected GPS

**RTK** real time kinematic (*corrections for GPS, similar to DGPS*)

**GPS/INS** GPS and inertial navigation system

**KF** Kalman filter

**GP** Gaussian process

**RTFM** read the fucking manual

**PCA** principal component analysis

# Glossary

**L<sup>A</sup>T<sub>E</sub>Xplugin:** A L<sup>A</sup>T<sub>E</sub>Xpackage.

**Term:** This term means stuff.

# Chapter 1

## Introduction

This thesis template should provide you with enough  $\LaTeX$  code to get writing immediately on your thesis, without having to learn too much  $\LaTeX$  up front. You may wish to save snippets from the template, or the entire template somewhere separate before you overwrite too much of it if you're unfamiliar with  $\LaTeX$ , so you can refer back to the source and the PDF it produces if you get stuck.

For basic  $\LaTeX$  usage and understanding, check out the  **$\LaTeX$  Book on Wikibooks**: <http://en.wikibooks.org/wiki/LaTeX> — this covers most of the  $\LaTeX$  basics, and even a few of the more commonly used packages in a lot more depth than this template.

For help on using individual  $\LaTeX$  packages, they typically have their own *excellent* documentation. You can find it on Google, of course, or on most installations (certainly TeXLive/MacTeX) you can type:

```
# texdoc hyperref
```

in a terminal/shell (may require Cygwin for use on Windows?), to bring up the manual for a specific  $\LaTeX$  plugin (in this example, the manual for the `hyperref` package).

If your problem isn't resolved through these 'read the fcuking manual ([RTFM](#)) methods', then try the excellent **TeX StackExchange site**: <http://tex.stackexchange>.

[com/](#)—this is a site where users ask and answer each others' questions, and is one of the best technical help sites around, thanks largely to the way answers are modified and moderated by the community. About 95% of the time, your question will already have been asked and answered, so search first! The rest of the time, you'll likely get some useful answers within 24 hours if your question is well asked (articulate and ideally with some example code or an image as to what you're after).

## 1.1 Structure of the Template

Where this document refers to file paths, it will be assumed the whole template is in a directory called `Thesis` so that, for example, the main `.tex` file is located at `Thesis/Thesis.tex` and the bibliography file at `Thesis/LaTeX/library.bib`.

The files in the thesis template are structured in various subdirectories:

- The main  $\text{\LaTeX}$  file is `Thesis.tex`, which sets up the document and includes the various supporting code and content.
- The `LaTeX` directory contains the supporting  $\text{\LaTeX}$  code for layout, packages, macros, etc, intended to simplify the writing process and keep the document style consistent.
- The *front matter*, which consists of everything from the title page up to (but not including) Chapter 1, is contained in the `FrontMatter` directory.
- The *Chapters* of the thesis are each contained in their own directory, named `ChapterABC` (where `ABC` is replaced with a short descriptive name for the chapter).
- The *Bibliography*, which is part of the 'back matter', is trivially produced by a few simple  $\text{\LaTeX}$  commands at the end of `thesis.tex`.
- The *Appendices*, also part of the 'back matter', are each contained similarly to the Chapters in their own directory, named `AppendixABC` (again replace `ABC` with a short name for the appendix).



The file structure of the template will start out something like the structure on the following page. Not all files are included (only one chapter, `ChapterPerception`, contains much) but there should be enough for you to get the general gist of it.

```

Thesis.....top level directory
├── Thesis.tex.....master .tex file
├── Thesis.tex.latexmain.....Vim-Latex-Suite project file
├── FrontMatter.....everything before Chapter 1
│   ├── FrontMatter.tex.....main .tex file, includes other front matter
│   ├── TitlePage.tex.....title page construction
│   ├── Declaration.tex.....standard declaration you need to sign
│   ├── Abstract.tex.....your abstract goes in here
│   ├── Acknowledgements.tex.....acknowledgements here
│   ├── Dedication.tex.....a dedication or something poignant and/or poetic
│   ├── Nomenclature.tex.....mathematical notation, abbreviations, acronyms
│   └── Glossary.tex.....definitions of specialised words used
├── ChapterIntro
│   └── Intro.tex
├── ChapterLiteratureReview
│   └── LiteratureReview.tex
├── ChapterPerception
│   ├── Perception.tex.....the main .tex file for the chapter
│   ├── Intro.tex.....example section
│   ├── Method.tex.....example section
│   ├── Results.tex.....example section
│   ├── Conclusions.tex.....example section
│   ├── .....put more sections for your chapter here
│   └── Figures.....directory for your figure files
│       ├── example.jpg.....an example figure from the template
│       ├── second-example.pdf.....another example figure
│       └── .....put more figures for your chapter here
├── ChapterEstimation
│   ├── Estimation.tex
│   └── Figures
├── ChapterExperiments
│   ├── Experiments.tex
│   └── Figures
├── ChapterConclusions
│   └── Conclusions.tex
├── AppendixSomething
│   └── Something.tex.....an example appendix
└── LaTeX.....contains supporting LATEX code
    ├── packages.tex.....included LATEX packages
    ├── layout.tex.....page layout settings
    ├── commands.tex.....some helper macros
    ├── dummy.jpg.....a dummy figure used if you don't have one ready
    └── library.bib.....BibTeX library

```

## 1.2 Personalising the Thesis

You can personalise the thesis template with your name, title, dates, etc. at the start of `Thesis/Thesis.tex`, where a bunch of macros define some text strings for re-use in the document. You generally won't need to delve into the `FrontMatter` directory, except where large chunks of one-off text are required (abstract, dedication, acknowledgements, glossary and nomenclature). Where possible, front matter content is auto-generated from the macros in `Thesis/Thesis.tex` or from the content of the thesis as you write it (i.e. tables/lists of contents/figures/etc).

## 1.3 Chapters and Appendices

You should come up with a theme or short name for each of your chapters. Of course, this will not necessarily happen on day one, nor will it necessarily stay the same throughout writing. There are six chapters in this template:

1. Introduction
2. Literature Review
3. Perception
4. Estimation
5. Experiments
6. Conclusion

### 1.3.1 Renaming Chapters

You'll want to rename the chapters and make your own  $\text{\LaTeX}$  file structure within each to make editing easier. What you *don't* want is a single `.tex` file for the whole document, or even any sizeable chapter, as this makes it very hard to navigate around the source code. You should generally consider splitting chapters up into one file per section, and you might want to split especially long sections into sub-sections. The

`\input{}` is used to achieve this, and acts similarly to the `#include` pre-processor directive from C/C++. The `\include{}` command is similar and will allow much the same inclusion though operates slightly differently.

When renaming a chapter or appendix, you'll need to complete the following steps. The only differences between a chapter and an appendix is whether they are included before (chapters) or after (appendices) the References section in the `Thesis/Thesis.tex` file and the prefix in the directory name (i.e. `Thesis/ChapterName` for chapters and `Thesis/AppendixName` for appendices).

1. Pick an unused chapter from the template to rename, or copy one of your existing chapters to a new folder in the thesis  $\text{\LaTeX}$  project directory. For this example, let's assume you're starting with the `Thesis/ChapterPerception` chapter directory, and will be writing a chapter on something relating to *Control*. Whatever the subject, try to keep the name short, distinctive, simple and *without spaces*.
2. Rename the chapter's directory from `ChapterPerception` to, say, `ChapterControl`
3. Rename the chapter's main `.tex` file (within the chapter directory you just renamed) to `Control.tex`.

The full path to the chapter's main `.tex` file should now be `Thesis/ChapterControl/Control.tex`.

4. Edit `Thesis/Thesis.tex` and update the `\include{}` statement to reflect the new name, e.g. change from this: `\include{ChapterPerception/Perception}` to this:

```
\include{ChapterControl/Control}
```

Do not include the `.tex` extension for `\include{}` or `\input{}`.

5. Edit `Thesis/ChapterControl/Control.tex` and modify the following:

- Change the `./ChapterIntro` macro to point to the chapter directory set in [Item 2](#) as a relative subdirectory of `Thesis`. For example:
- Change the `\label{}` after the `\chapter{title...}` statement to select a new cross-referencing label for the chapter (this is used within the  $\LaTeX$  source when you want to refer to something within that chapter, for example “[Chapter 3](#) shows that...”), e.g. from

```
\label{ch:perception}
```

to

```
\label{ch:control}
```

- Change the `\input{}` statements by updating the paths to the chapter directory chosen in [Item 2](#), e.g. from

```
\input{ChapterPerception/Intro}
```

to

```
\input{ChapterControl/Intro}
```

At this point you may also wish to create new section files, or re-use (and modify) old ones.

6. Make sure you update any renamed/moved files in your  $\LaTeX$  project (if you’re using an IDE-like editor), or ensure new files points to the master `.tex` file or directory. How this is done will depend on your editor, and whether it even has a concept of projects or master  $\LaTeX$  files. Depending on how your editor/IDE works, it may be useful to name section `.tex` files with the prefix `Section` (e.g. `SectionDiscussion.tex`).
7. You can now go about fixing up chapter and section titles, and modifying the content of your new chapter appropriately.

### 1.3.2 Re-ordering Chapters

Changing the order of chapters or appendices is simply a matter of changing the order they are listed in `Thesis/Thesis.tex` in the `\include{}` statements. Furthermore,

converting one to the other is technically just a matter of moving its `\include{}` statement either before (for chapters) or after (for appendices) the `\appendix` command — you should also follow the steps for renaming a chapter to change between `Chapter...` and `Appendix...` if you do so in order to avoid confusion.

## 1.4 Single- or Double-Sided Page Printing?

The template should work just fine for either single- or double-sided page printing, though depending on your binding method you may need to increase the inside margin. In `Thesis/Thesis.tex` you can provide the `\documentclass` function with either `oneside` or `twoside` to choose between the two.

## 1.5 Organisation of the Thesis Template

The thesis template contains general instructions on usage in this chapter, as well as some example chapters to demonstrate basic  $\text{\LaTeX}$  usage for the types of features others at the ACFR have found useful in the past. There's always going to be many ways to skin the same cat, and even though  $\text{\LaTeX}$  plugins are very stable, there will be changes over the course of a few years, with newer, better solutions coming along.

**Chapter 2** is an example literature review chapter, demonstrating the basics of referencing.

**Chapter 3** demonstrates some basic features of  $\text{\LaTeX}$ , like equations, figures, sub-figures, tables, etc. These are the things that are pretty straight forward, and can be done with the most commonly used packages.

**Chapter 4** provides examples of some less-often used features, like theorems, algorithms, protocol diagrams, etc. These examples generally require a specific package, and there's often a few different ones to choose from depending on exactly what you're after. Quite often, you'll be forced to choose between the various packages, as they may be incompatible.

**Chapter 5** contains the quirkiest of examples; things that are hard to do, but unlikely to be very useful except to a few people. If you're adding to this template, feel free to do so here (provided you don't break any other packages!), and if you find something in there especially useful, you could clean it up, and modify the template source to move it to **Chapter 4** instead.

**Chapter 6** doesn't really provide much, it's just there because you'll want a conclusion.

The **Bibliography** goes *after* the last chapter and *before* the first appendix. It is automatically generated from citations (see **Chapter 2** for citation methods).

**Appendix A** shows an example of source code listing and especially large tables (multi-page and landscape); the sorts of things you may want in an appendix.

# Chapter 2

## Literature Review

Firstly, in order to make citations, you're going to need a BibTeX file. There's one in the template, but it's not going to include the references you need, so delete it and make your own. You can make them by hand, but that sucks, so get a program like Mendeley (Win, Mac, Linux), JabRef (Win, Mac, Linux), Papers (Win, Mac) or BibDesk (Mac only). If you're already using something like EndNote, you can export to any of the aforementioned library systems (which actually support BibTeX) very easily, just have a look on the website for whichever program you choose for a step by step guide.

Mendeley is a reference manager and academic social network that will automatically keep a `library.bib` file up to date for you, Papers will let you export your Papers library to a `.bib` file, and JabRef and BibDesk directly edit `.bib` files, so take your pick as to how you want to generate it.

### 2.1 A Bit on Citations

You should have a look at the source code here for information on how to cite<sup>1</sup>. First I'll cite some work using the `\cite` command, like [Connolly and Scheduling \(2008\)](#)

---

<sup>1</sup>you'll find it all in the `Thesis/ChapterLiteratureReview/LiteratureReview.tex` file



and Williams et al. (2006). This just uses the default method set up when the `natbib` package was included, so you may wish to change those options, referring to `Thesis/LaTeX/packages.tex` and the `natbib` documentation. The template has been set up to use square brackets, numbers, alphabetically-sorted bibliography, and automatic sorting and compressing of references where multiple are cited at once.

You should read the `natbib` documentation, but for a quick start, you can do most of your citing as follows:

- For basic citations, `\cite{label}` gives Underwood et al. (2010).
- For named or ‘text’ citations, `\citet{label}` gives Underwood et al. (2010).
- For a full author list, add an asterisk, so `\citet*{label}` gives Underwood, Hill, Peynot, and Scheduling (2010).

Things change a bit if you change the options for `natbib`, letting you choose other styles like (Author, year) or Author (year), etc.

For components of citations, you can use

- `\citeauthor` gives Underwood et al.
- `\citeauthor*` gives Underwood, Hill, Peynot, and Scheduling.
- `\citeyear` gives 2010.
- `\citeyearpar` gives (2010).

Sometimes you might want to cite a bunch of papers in a list, which would look something along the lines of Allen et al. (2009); Johnson-Roberson et al. (2010); Mahon and Williams (2003); Merry (2009); Nettleton et al. (2006); Stroustrup (2000); Underwood et al. (2010); Williams et al. (2006). Notice how  $\LaTeX$  has simplified the list of citation numbers into ranges, rather than listing every single one<sup>2</sup>.

Finally, you might want to point to a specific chapter or section of a book or paper, for example using `\cite[note]{citekey}` to give, for example, (Underwood et al., 2010, Section 4) or (Nettleton et al., 2006, Chapter 3).

---

<sup>2</sup>Note that if IEEE-style references are selected in `Thesis.tex` they’re numbered in the order that they appear in the bibliography, which is currently set to sort alphabetically by author, then by age. If you prefer to list references by age, remove the `sort` option in `packages.tex`.

You'll note in all cases that you can click on the author name or year number and it'll take you to the Bibliography entry for that paper (thanks to the `hyperref` package). You can fairly easily check for any possible duplication (if you've got multiple `citekeys` pointing to the same paper) by reading through your Bibliography which, for now at least, is in alphabetical order by first author.

You might want to double check the formatting in the Bibliography is correct. It's very easy to chop and change between styles or modify the style to be precisely what you're after. For example, `plainnat` tends to include the URL key, which some bibliography programs will automatically include, and you may or may not want in your output. You can either modify `plainnat` or use the `acfrplainnat` style provided. This can be changed in `Thesis/Thesis.tex` by modifying the call to `\bibliographystyle{}`. There are a multitude of styles available, so you probably don't need to customise it yourself, especially if you're aiming to match a particular journal.

Here are some gratuitous references to show how the bibliography style formats the various entry types:

- an Article: `Dahiya et al. (2010)` by `Dahiya et al.` in `2010`;
- a Book: `Stroustrup (2000)` by `Stroustrup` in `2000`;
- a Booklet: `Australian Academy of Science (2010)` by `Australian Academy of Science` in `2010`;
- an Inbook: `Lynch (1996)` by `Lynch` in `1996`;
- an Inproceedings: `Bennewitz et al. (2002)` by `Bennewitz et al.` in `2002`;
- a Mastersthesis: `Merry (2009)` by `Merry` in `2009`;
- a Misc: `Beebe (2002)` by `Beebe` in `2002`;
- a Phdthesis: `Douillard (2009)` by `Douillard` in `2009`;
- a Proceedings: `Rye and Furukawa (2001)` by `Rye and Furukawa` in `2001`;
- a Standard: `AS4 (2009)` by `AS4` in `2009`;
- a Techreport `Likhachev et al. (2005)` by `Likhachev et al.` in `2005` and
- an Unpublished `Hill (2011)` by `Hill` in `2011`.

## 2.2 Cross-Referencing Within Your Document

You can reference chapters, sections, figures, examples, etc. very easily, thanks to the `hyperref` package (which includes the `\autoref` command). While  $\LaTeX$  already provides the `\ref` command, there are advantages to using `hyperref` and its `\autoref` command.

First, the thing you want to reference needs to have a label, such as the label for this section, which has been specified by inserting: `\label{sec:crossrefs}`. For convenience, you might want to use a prefix like ‘sec’ for sections, ‘ch’ for chapters, ‘fig’ for figures, etc. This may make auto-complete (if your editor supports it) for these labels easier, and will reduce the chance of collisions between labels where you have a figure, a table and a section of the document all about much the same material.

If your editor doesn’t nicely handle auto-completion of labels, you might want to check out the `showkeys` package, which lets you display the cross-referencing labels in your PDF. There’s a few lines in `Thesis/LaTeX/packages.tex` that you can uncomment to show labels in your PDF (but don’t forget to remove them before printing the final doc ;).

Once you have a label, you can then generate references to it with `\ref`, `\autoref` or `\hyperref` as follows:

- `Section~\ref{sec:crossrefs}` → “Section 2.2”
- `\autoref{sec:crossrefs}` → “Section 2.2”
- `\hyperref[sec:crossrefs]{This section}` → “This section”

Note which parts are *clickable PDF links* (only the number for `\ref`, but the whole thing for the other two).

Also note that only `\hyperref` is able to reference unnumbered sections/figures/etc. correctly (since the other two rely on the item’s number as part of the reference), for example correctly linking to the [Nomenclature Section](#) in the front matter, or making nice links to multiple references, such as referring to “[Sections 2.1 and 2.2](#)” (`\autoref` doesn’t support plurals at the time of writing).

The `hyperref` package documentation provides more info on many more linking/referencing commands, for example, referencing items by their page number (e.g. “... as seen on [page 13](#)”) or URLs (e.g. “... as seen on [the ACFR website](#)”).

# Chapter 3

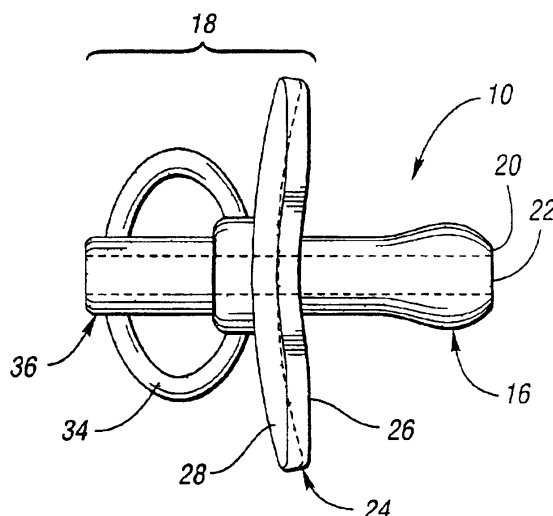
## Example Chapter on ‘Perception’ Stuff

This chapter provides some examples of the basic features of L<sup>A</sup>T<sub>E</sub>X. I won't bother including much L<sup>A</sup>T<sub>E</sub>X code in the PDF, since it gets a bit more complex from here-on, so refer back to the `.tex` files for info.

### 3.1 Figures

A basic figure can be seen in [Figure 3.1](#). Note that L<sup>A</sup>T<sub>E</sub>X will arrange figures (and other floating environments) as it sees fit, so that they may not be exactly where you expect them to be, but they will be close by. You should generally refer to figures in the text using `\autoref` rather than by their position on a page, otherwise it'll get confusing later on.

**Don't try to get the layout of figures, or any other floating environments ‘just right’ until your document is 100% complete.** It's a waste of time, as things will change as you add/remove stuff elsewhere in the document. If it's truly vital that a floating environment is immediately after a specific piece of text, you can do this, but otherwise just let L<sup>A</sup>T<sub>E</sub>X sort it out.

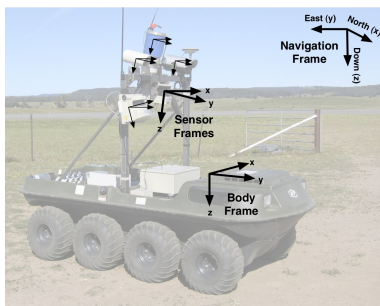


**Figure 3.1** – This is the first example figure. Note that the caption here is different to the one in the table of contents. Note also that I’ve inserted it with the `\fig` macro that’s in `Thesis/LaTeX/commands.tex`. Finally, there is actually no image file for this figure, so it’s been substituted for the dummy figure by the `\fig` macro. This is quite useful for getting a feel for the layout, before you’ve necessarily got a figure ready for placement.

The `\fig` macro will automatically pick up PDF, JPG and PNG images when generating a PDF with `pdflatex`, and EPS images when generating a DVI file (though the latter has not been well tested with the rest of the template), and no file extension needs to be specified when using the `\fig` macro. For example, [Figure 3.2](#) is a JPEG image, and [Figure 3.3](#) is a vector-art PDF.

In general, the following tips for making/placing figures will give you high quality images without making the final `Thesis.pdf` file unnecessarily large.

For **photographic or scanned images**, use the JPEG format and always convert *from the original image* rather than resizing, drawing over it, resizing, etc (look up “JPEG Rot” for more info). Your final image should be about 150–300 dpi for printing, so if it’s the full width of the text (i.e. 6 inches), it should be between 900 and 1800 pixels wide (about 2 megapixels assuming it’s a standard landscape photograph). There’s no realistic benefits to using the original 10-megapixel image from

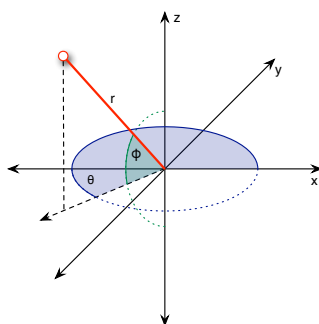


**Figure 3.2** – Some coordinate frame definitions shown an 8x8 Argo vehicle.

your camera; scale it down, or you won't be able to e-mail it to your supervisor to review.

For **line art, diagrams, computer drawings, etc**, ideally you want to draw them in a vector-art program and convert to PDF, however a PNG file will generally be okay as well. Vector art has the possible advantage of the reader being able to search for or select any text in the diagram in the final PDF. The PDF format can handle both raster and vector graphics, so if you're drawing over a photo or screenshot, you should consider whether you should use a JPEG or PNG raster format for certain elements of the PDF.

For **screenshots, complex vector art diagrams or complex datasets**, it's generally best to generate a PNG file. Don't use JPEGs for this, they produce artefacts



**Figure 3.3** – An example of measurement in a circular polar coordinates on a Cartesian frame.

(strange blurring and colouring) near hard edges as the compression algorithm is not designed for clean hard edges. You may even want to convert your vector-art diagram/dataset to a PNG if there's a lot of elements (e.g. 1,000+) as complex vector graphics can be slower to render and have a larger file size than a PNG. You should aim for about 300–600 dpi for a PNG diagram, especially if there are any curved shapes, shadows, etc.

If you're unsure how big you'll print your raster-based image in the final thesis, the text width is about 6 inches wide, so set your image width to:

- 900 pixels for 150 dpi;
- 1800 pixels for 300 dpi; or
- 3600 pixels for 600 dpi.

There's not a big difference in file size, and it'll save you time if you resize images later on.

### 3.1.1 Figure Placement

You can achieve figure placement with varying degrees of success/reliability. Figure placement is largely automatic, but you can direct  $\LaTeX$  with a placement argument for float environments (the first argument for the `\fig` macro). Generally, the placement argument can be any combination of **h** (here), **t** (top of page), **b** (bottom of page) and **p** (whole page), in order to 'ask  $\LaTeX$  nicely' to place your figure that way. These are attempted in the order you specify, and refer to any nearby page (not necessarily the page you expect!). I usually just use `htbp` and let  $\LaTeX$  sort it out, and find it's fine about 95% of the time. For especially large pictures, it's usually best to go with just **p** and, if necessary, use a `\clearpage` command to force  $\LaTeX$  to finish placing any unplaced floating environments at that point in the document.

If you really just want to place a figure **here**, i.e. with guaranteed placement with respect to text (rather than let it float to a 'nice' position), you should change the float 'placement' argument to **H**. This isn't compatible with the others, since it's forced rather than requested.



## 3.2 Tables

Tables can be pretty messy, and there’s a million features in a lot of  $\text{\LaTeX}$  packages, which you’ll learn as you need specific things.

The best advice for now is probably to check out the `booktabs` package, which makes tables look much nicer than the ‘usual’ way of making tables in  $\text{\LaTeX}$ . They have a few guidelines/rules, like “don’t use vertical lines”, and simplify some of the process of putting the horizontal lines in tables.

**Table 3.1** – A simple example of a table to describe something

Label	Value	Description
Width	3 cm	The width of the object
Height	8 cm	The height of the object
Depth	14 cm	The depth of the object
Shape	ovoid	The geometric shape descriptor of the object

In this case, I’ve forced the table placement using the `H` placement argument. This is perhaps a bit more common for simple tables where maybe you just want to make a brief point in a slightly more graphically structured way.

In [Table 3.2](#), some of the nice features of the `tabularx` package are demonstrated, which lets you set properties on a column, and simplifies text-wrapping combined with auto-width calculations. This table is also placed using `htb` rather than `H`, so is properly floating—note the effect on the line spacing (the rest of the document has a longer line spacing, which doesn’t infiltrate floats).

**Table 3.2** – A slightly more complex example table, this time using features from the `tabularx` package.

<b>Bold Text Wrap and Fill Width</b>	
<b>Note</b>	It's much easier to have a column that fills the width of the table and also provides text-wrapping using the <code>tabularx</code> package and the <code>X</code> column specifier, than it is with the regular <code>p</code> column specifier.
<b>Also Note</b>	It's also easy to make elements in a column automatically bold with <code>tabularx</code> , without having to specify them as bold on each line.

### 3.3 Equations

For many people, equations are one of the best things about  $\text{\LaTeX}$ . They're laid out well and are reasonably easy to enter, and can even be generated programmatically, e.g. from MATLAB and its symbolic toolbox (which cannot be said of most publishing software).

For basic in-line numerical or equation elements, you just wrap the text in `$` symbols. For example we could point out that  $1 + 1 = 2$  or that  $e^{i\pi} - 1 = 0$ . If you want to number the equation (for referencing) or it's at all complex, try the `equation` environment.

$$1 + 1 = 2 \tag{3.1}$$

Of course you can cross-reference equations just like any other object using a label, letting you refer, for example, to [Equation 3.1](#) or just [\(3.1\)](#). For something a bit more interesting, check out Erwin Schrödinger's famous equation in [Equation 3.2](#).

$$i\hbar \frac{\partial \psi}{\partial t} = \frac{\hbar^2}{2m} \nabla^2 \psi + V(\mathbf{r})\psi \tag{3.2}$$

The `amsmath` toolbox also provides a lot of useful equation environments that let you structure the layout of equations in various nice ways. Read the documentation for lots of very good info, but here's an example or two.

Aligned equations (with optional equation numbering):

$$x = \cos^2(\theta) + \sin^2(\theta) \quad (3.3)$$

$$\theta = \frac{\pi}{4} \quad (3.4)$$

$$\begin{aligned} y &= x^2 + 5x - 3 \\ &= 3 \end{aligned} \quad (3.5)$$

Multi-line equations:

$$\begin{aligned} x &= 3 \sin(\omega) + 2 \sin(\phi) + a \sin(\omega) + \sin(\beta) + 2 \sin(\gamma) \\ &\quad + 6 \cos(\phi) - 7 \cos(\omega) + \cos(\beta) + \cos(\gamma) \\ &\quad - 2 \tan(\alpha) + 3 \tan(\pi) - 7 \tan(\phi) + 3x \end{aligned} \quad (3.6)$$

Comments in lists of equations and specific horizontal spacing:

$$x = y^2 - 8x + 63 \quad \text{from the problem} \quad (3.7)$$

$$9x = y^2 + 63$$

$$x = \frac{3^2 + 63}{9} \quad \text{substitute } y = 3 \text{ from (3.5)}$$

$$x = 8 \quad (3.8)$$

You can do pretty much anything. Google is your friend (but you may need to know the actual names of some symbols you use often!). Documentation may move or go out of date, but you could try looking here: <ftp://tug.ctan.org/pub/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>. Otherwise just search for help on “AMS Math LaTeX”, “Comprehensive LaTeX Symbol List” and “Comprehensive LaTeX Math Symbols”.

### 3.3.1 Learning to Set Equations

Complicated equations can be a bit daunting when you are starting out with  $\text{\LaTeX}$ . A nice resource for learning to set equations, or for creating complicated equations, is the CODECOGS equation editor, at <http://www.codecogs.com/latex/eqneditor.php>. This editor has many drop down menus for selecting equation structures, which are parsed and displayed as you type. Once your equation is correct, the  $\text{\LaTeX}$  code can be pasted into your document.

## 3.4 Acronyms

You can use acronyms in your thesis, and have them automatically expanded on first use but abbreviated on subsequent uses. This is done using the `acronym` package. If you want the acronyms to be listed in the Nomenclature, then they should be defined there using the `\acro` macro. Any macro defined in the Nomenclature is available for use in the document, but *only those used in the document* will appear in the acronyms table in the Nomenclature<sup>1</sup>.

Any acronym used in the document will be automatically expanded on first use, and be abbreviated on subsequent uses, unless you specify expanded/contracted versions. For example, this template was created for use by students at the Australian Centre for Field Robotics (**ACFR**). In the past, there have been several templates used by students at the **ACFR**, with varying degrees of success. It was decided that the **ACFR** should have a single common template to make this easier for students new to  $\text{\LaTeX}$ , and you are now using it.

But I may want to *pluralise* the acronym like so: Kalman filters (**KFs**) or Gaussian processes (**GPs**). Or I may want to refer to it by its *long* name without even showing it's an acronym: principal component analysis. Or maybe repeat it in *full*, because

---

<sup>1</sup>You can use the `\acused` macro in the Nomenclature (after defining the acronym) to prevent an acronym ever expanding in the text; this is useful for especially common acronyms, which can be looked up in the Nomenclature, but would not ever need to be written out in full in the text.

I think the reader probably didn't bother reading where I first defined it: Kalman filter ([KF](#)). And now, just to show some of the other features, I'm going to use a few of the acronyms here:

- real time kinematic ([RTK](#))
- [INS](#)<sup>2</sup>
- [GPS](#) and inertial navigation system ([GPS/INS](#))<sup>3</sup>
- differentially-corrected [GPS](#) ([DGPS](#))<sup>4</sup>

There are options and macros for the `acronym` package that allow resetting the 'used' state of acronyms at various points in the document, for example to allow you to have the first use of each acronym *in each chapter* spelt out in full.

For example, if I reset *all* acronyms right here using the `\acresetall` macro... the next reference to an acronym will be spelt out in full: Australian Centre for Field Robotics ([ACFR](#)), and subsequent usage of the acronym will be abbreviated: [ACFR](#).

## 3.5 Summary

[Chapter 4](#) and [Chapter 5](#) take things to a more advanced level.

---

<sup>2</sup>This one was specified with `\acused`.

<sup>3</sup>This one redefines the short version to add the '/?.

<sup>4</sup>This one refers to another acronym within its long version.

# Chapter 4

## Some Estimation Stuff

The examples in this chapter may be a little less commonly used than those in [Chapter 3](#).

### 4.1 Theorems and Hypotheses

You can define simple ‘theorem’ environments using the `amsthm` and `thmtools` packages. These could be used for presenting short statements of theorems, examples, hypotheses, notable outcomes, etc. Reasons to use this environment for such things would be:

- The ability to list them in a Table of Theorems or List of Examples, etc at the start of the thesis.
- The ability to refer to them using `\autoref` just like you can with figures/tables/sections/etc.
- Keeping them in a standard format, without having to specify any of the formatting yourself.

The ‘Hypothesis’ type of theorem environment has been set up in `Thesis/LaTeX/commands.tex`, which provides the `hypothesis` environment to do something like the following:

**Hypothesis 4.1** (Descartes). *I think, therefore I am.*

Which you can refer to as [Hypothesis 4.1](#) in the text with all the usual benefits.

There's also a `theorem` environment set up, so you could provide some theorems:

**Theorem 4.1** (Zeroth Law of Thermodynamics). *If system A and system B are in thermal equilibrium with system C, then system A is in thermal equilibrium with system B*

**Theorem 4.2** (First Law of Thermodynamics). *Energy can be neither created nor destroyed. It can only change form.*

**Theorem 4.3** (Second Law of Thermodynamics). *Entropy... it only goes up.*

**Theorem 4.4** (Third Law of Thermodynamics). *The entropy of a perfect crystal at absolute zero is exactly equal to zero.*

Note the numbering is independent for each type (i.e. [Hypothesis 4.1](#) vs [Theorem 4.1](#)). Also, these two environments are just the basic defaults; you can change their appearance to make them stand out a bit more or differentiate between them in a more visual way. See the `thmtools` documentation for info on that.

You should also look at the “List of Theorems” in the front matter of this template, which includes *all* theorems from the `thmtools/amsthm` packages, labelling them with the theorem ‘type’. This looks a bit weird if you use theorems for very different purposes (listing, for example, ‘examples’ along-side ‘theorems’ and ‘significant contributions’), but if you’re only using one or two (or can perhaps work out how to separate them out), it’s quite useful. If you don’t like the list, you can also just remove it entirely from `Thesis/FrontMatter/FrontMatter.tex`.

If you want something framed, rather than just a slightly different text environment, you might want to look in [Section 4.2](#), which makes a much more complex theorem environment for showing examples.

Joop’s example of proposition use:

$$\begin{aligned}
& \text{Maximise} && \sum_{(s,t) \in \mathcal{E}} c_{st}^T y_{st} \\
& \text{Subject to} && A_{st} y_{st} - A_{su} y_{su} = 0 \quad \forall s \in \mathcal{V}, t \in \mathcal{N}(s), \forall u \in \mathcal{N}(s) \setminus t \\
& && \sum_k y_{st}^k = 1 \quad \forall (s,t) \in \mathcal{E} \\
& && 0 \leq y_{st}^k \leq 1 \quad \forall (s,t) \in \mathcal{E}, k = 1, \dots, |\mathcal{A}_s| |\mathcal{A}_t|.
\end{aligned} \tag{4.1}$$

Some text here omitted for brevity ...

### 4.1.1 Solution Consistency

The LP of [Equation 4.1](#) is defined over the edges; as such, constraints are required to ensure the solution of the LP remains consistent in the node variables.

**Definition 4.1** (Marginal). *Let  $m_{s|t} = A_{st} y_{st}$  be a marginal, for node variable  $x_s$ , of the edge variable  $y_{st}$ .*

**Proposition 4.1** (Solution Consistency). *The solution for the edge variables  $\{y_{st} | \forall t \in \mathcal{N}(s)\}$  is consistent in the node variable  $x_s$  when the marginals  $\{m_{s|t} | \forall t \in \mathcal{N}(s)\}$  are all equal.*

*Proof.* The proof can be obtained by substitution of  $y_{st}^k = x_s^i x_t^j$  and  $\sum_i x_s^i = 1$ .  $\square$

Consistency constraints are specified over pairs of edges, i.e. as the difference between pairs of marginals  $m_{s|t}$  and  $m_{s|u}$ . For a given node  $s$  one edge is used as the reference edge; edge  $(s, t)$  in [Equation 4.1](#). All consistency constraints are specified relative to the reference edge resulting in a minimum of constraints generated. Subsequently solving [Equation 4.1](#) will result in a solution for the edge variables. The mapping from  $y_{st}$  to the node variables  $x_s$  is given by the following proposition.

**Proposition 4.2** (Solution Mapping). *If the linear program of [Equation 4.1](#) has a feasible solution, then the mapping from  $y_{st}$  to  $x_s$  is given by  $x_s = m_{s|t}$  for any  $t \in \mathcal{N}(s)$ .*



*Proof.* The equality  $x_s = m_{s|t}$  follows directly from the definitions of  $A_{st}$ ,  $y_{st}$  and  $m_{s|t}$ . [Proposition 4.1](#) permits any  $t \in \mathcal{N}(s)$  provided the solution is consistent. Solution consistency, and therefore [Proposition 4.1](#), is ensured by virtue of a feasible solution; all constraints are met.  $\square$

## 4.2 Examples

I've used the `mdframed` package to make a simple framed-box example environment, in conjunction with the `thmtools` and `amsthm` packages. You use it much the same as the theorems and hypotheses shown in [Section 4.1](#), but it's visually quite different.

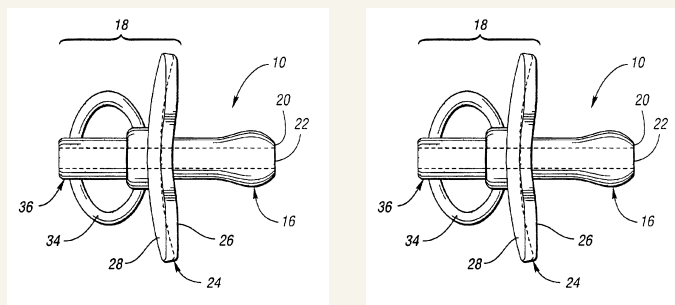
It's simple in that it's easy to create, though it's a little complex behind the scenes (i.e. what's in `Thesis/LaTeX/commands.tex`), and will sometimes have odd vertical spacing that requires adjustment afterwards (much like placement of floating environments, just leave the spacing as is and fix it just before you print the final document if need be — it's usually not too bad).

The `example` environment implemented in this template can span multiple pages, though it tends to look better when it doesn't, especially if there are large objects in the environment which leave big white spaces. So in this way it's not like a standard floating environment, but it's also not meant to be, and you'll be a bit disappointed if that's what you're expecting.

You can see an example example environment in [Example 4.1](#). The usage doesn't change a lot, but it looks very different.

### **Example 4.1 — A Basic Demonstration**

This is an example of the example environment. You can put most things in here, including figures, tables, equations. Usually it's a bit tight on space if you want to put more than one in, so here's two side-by-side:



(a) A picture of a dummy

(b) A very similar dummy

**Figure 4.1** – You can put an overall caption here for both sub-figures, individually referencing (a) and (b) if need be.

Sorry, I've not made a nice macro for sub-figures like `\fig`, so it's a bit messier.

## 4.3 Algorithms

The `algorithm2e` package provides one implementation of the algorithm environment, however there are other packages (like `algorithm`) which are probably also worth looking into. *I* found this one to be easier to do what *I* wanted; you may or may not.

A simple example is provided below, and as always you can cross-reference [Algorithm 4.1](#) (thanks again to the `hyperref` package). There's also a List of Algorithms provided in the Front Matter.

## 4.4 Summary

---

**Algorithm 4.1:** Cost Cell Expansion

---

**input** : A cost map as a 2D array of cells

**output:** An expanded cost map as a 2D array of cells

**foreach** cell  $\in$  *all map cells* **do**

    maxCost  $\leftarrow$  CostOfCell(cell);

    //determine max cost in 'nearby' cells using kernel

**foreach** kernelCell  $\in$  *all kernel cells relative to cell* **do**

        maxCost  $\leftarrow$  Max(maxCost, CostOfCell(kernelCell));

    outputCell  $\leftarrow$  maxCost;

---

# Chapter 5

## Experiments That Prove Things

Not sure what to put in here just yet, so it's just a fairly empty set of sections.

### 5.1 Example Section

This section has some stuff in it that's about a thing.

### 5.2 Example Section Two

This section has some stuff in it that's about a thing, even better than [Section 5.1](#).

Woo!

### 5.3 Third Example Section

This section has some stuff in it that's about a thing like that in [Section 5.2](#) but isn't quite as exciting. Sorry.

## 5.4 Example Section Four

This section has some stuff in it that's about a thing that is way more exciting than anything else in [Chapter 5](#)

## 5.5 Summary

“Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?”

(Section 1.10.32 of “de Finibus Bonorum et Malorum”, written by Cicero in 45 BC)

1914 translation by H. Rackham

“But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness. No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful. Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toil and pain can procure him some great pleasure. To take a trivial example, which of us ever undertakes laborious physical

exercise, except to obtain some advantage from it? But who has any right to find fault with a man who chooses to enjoy a pleasure that has no annoying consequences, or one who avoids a pain that produces no resultant pleasure?"

“At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.”

(Section 1.10.33 of “de Finibus Bonorum et Malorum”)

1914 translation by H. Rackham:

“On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame belongs to those who fail in their duty through weakness of will, which is the same as saying through shrinking from toil and pain. These cases are perfectly simple and easy to distinguish. In a free hour, when our power of choice is untrammelled and when nothing prevents our being able to do what we like best, every pleasure is to be welcomed and every pain avoided. But in certain circumstances and owing to the claims of duty or the obligations of business it will frequently occur that pleasures have to be repudiated and annoyances accepted. The wise man therefore always holds in these matters to this principle of selection: he rejects pleasures to secure other greater pleasures, or else he endures pains to avoid worse pains.”

# Chapter 6

## Conclusion

The purpose of this thesis is to say stuff. It really did say a lot of stuff. I hope you liked it.

This chapter provides a summary of the content in the thesis, a list of contributions to the field, and a discussion of potential future work.

### 6.1 Summary

This thesis...

In [Chapter 2](#), current work in the field is ...

In [Chapter 3](#), the issues of ...

[Chapter 4](#) demonstrates ...

A series of experiments were shown in [Chapter 5](#), to validate the work in [Chapter 3](#) and [Chapter 4](#) ...

### 6.2 Contributions

This thesis extends previous work ...

Specific contributions in detail are as follows:

- A contribution
- Another contribution
- Yet more contributions ...

## 6.3 Future Work

As has been outlined in [Chapter 2](#), there is a need to further improve stuff, which was done in [Chapters 3](#), [4](#) and [5](#) within the scope of this thesis. Some specific directions of work which have been highlighted by the analyses in this thesis as areas for further improvement are ...



# List of References

- AS 4024.3301-2009: Safety of machinery - robots for industrial environments - safety requirements. May 2009. ISBN 0-7337-9152-2.
- Tom Allen, Andrew Hill, James Underwood, and Steven Scheduling. Dynamic path planning with multi-agent data fusion: the parallel hierarchical replanner. In *Proceedings of the IEEE 2009 International Conference on Robotics and Automation*, pages 4445–4450. IEEE Press, May 2009. ISBN 978-1-4244-2788-8.
- Australian Academy of Science. The science of climate change: Questions and answers., August 2010.
- Nelson H. F. Beebe. A bibliography of literate programming.  
<http://www.literateprogramming.com/litprog-bib.pdf>. March 2002. Version 2.49, accessed 17 July 2001.
- M. Bennewitz, W. Burgard, and S. Thrun. Learning motion patterns of persons for mobile service robots. In *Proceedings of the IEEE 2002 International Conference on Robotics and Automation*, volume 4, pages 3601–3606, 2002.
- Laura A. Connolly and Steven Scheduling. Comparative analysis of sensors, algorithms, and models within a navigation system. *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 26–32, August 2008.
- R. S. Dahiya, G. Metta, M. Valle, and G. Sandini. Tactile sensing—from humans to humanoids. *IEEE Transactions on Robotics*, 26(1):1–20, 2010.
- Bertrand Douillard. *Laser and vision based classification in urban environments*. PhD thesis, The University of Sydney, January 2009.
- Andrew Hill. Personal communication, June 2011.
- Matthew Johnson-Roberson, Oscar Pizarro, Stefan B. Williams, and Ian Mahon. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51, 2010.

- Maxim Likhachev, David Ferguson, Geoffrey Gordon, Anthony (Tony) Stentz, and Sebastian Thrun. Anytime dynamic A\*: the proofs. Technical Report CMU-RI-TR-05-12, Robotics Institute, Carnegie Mellon University, May 2005. URL [http://www.ri.cmu.edu/pub\\_files/pub4/likhachev\\_maxim\\_2005\\_2/likhachev\\_maxim\\_2005\\_2.pdf](http://www.ri.cmu.edu/pub_files/pub4/likhachev_maxim_2005_2/likhachev_maxim_2005_2.pdf). Accessed 17 July 2011.
- Nancy A. Lynch. *Distributed Algorithms*, chapter 13: Atomic objects, pages 397–454. Morgan Kaufmann Publishers, 1996.
- Ian Mahon and Stefan Williams. Three-dimensional robotic mapping. In *Proceedings of the 2003 Australasian Conference on Robotics and Automation*. Australian Robotics and Automation Association, Sydney, Australia, 2003.
- Laura Merry. Generic architecture design for navigation filtering. MPhil thesis, The University of Sydney, August 2009.
- E. Nettleton, S. Thrun, H. Durrant-Whyte, and S. Sukkarieh. Decentralised SLAM with low-bandwidth communication for teams of vehicles. In Shinichi Yuta, Hajima Asama, Erwin Prassler, Takashi Tsubouchi, and Sebastian Thrun, editors, *Field and Service Robotics*, volume 24, pages 179–188, 2006.
- D.C Rye and T. Furukawa, editors. *Proceedings of the 2001 Australian Conference on Robotics and Automation*, 2001. Australian Robotics and Automation Association, Sydney, Australia.
- Bjarne Stroustrup. *The C++ Programming Language*, volume 1 of *The Series*. Addison Wesley, The address., 2nd edition, Jan. 2000. This is a note.
- J.P. Underwood, Andrew Hill, Thierry Peynot, and S.J. Scheduling. Error modeling and calibration of exteroceptive sensors for accurate mapping applications. *Journal of Field Robotics*, 27(1):2–20, 2010.
- S.B. Williams, G. Dissanayake, and H. Durrant-Whyte. An efficient approach to the simultaneous localisation and mapping problem. *Proceedings of the IEEE 2006 International Conference on Robotics and Automation*, pages 406–411, 2006.

# Appendix A

## An Example Appendix

As an appendix, this should contain some content that's not really required for the argument in the main body of the thesis, but is clearly relevant and supports the work.

### A.1 Code Listings

The `listings` package allows you to include code listings or other formatted text with some parsing to make them more readable than simply calling `\input{}` on the code file.

**Listing A.1** – MATLAB script for interactive radians to degrees converter

```
% A short script to help the user convert values in radians into degrees
% Numerical values will be converted, enter a blank input to exit
while(1),
    % Ask the user for a value
    rads = input('Enter a value in radians (enter nothing to exit): ');
    % If its blank, exit
    if isempty(rads)
        continue;
    end
    % Otherwise convert it to degrees and tell them the answer
    degs = rads * 180/pi;
    disp( [num2str(rads) ' radians is ' num2str(degs) ' degrees.' ] );
```

```
end
disp('Goodbye.');
```

A number of languages are supported with basic syntax highlighting and formatting.

## A.2 Multi-Page Tables

The `supertabular` package allows tables to span multiple pages using the `supertabular` environment (in place of `tabular`). This has already been used in the [Nomenclature Section](#) in the front matter, allowing the notation to span multiple pages if necessary. [Table A.1](#) shows an example of a table spanning two pages. Note that such tables are no longer floating elements (i.e. there's no `table` environment anymore), and the header/footer for the whole table, and ones repeated on each new page, can be defined through `supertabular` macros rather than as part of the table to copy headers across each page.

**Table A.1** – This table is especially long, so it's been turned into a `supertabular` environment allowing it to span multiple pages.

<b>first × second = RHS</b>			
1	×	1	= 1
1	×	2	= 2
1	×	3	= 3
1	×	4	= 4
1	×	5	= 5
1	×	6	= 6
1	×	7	= 7
1	×	8	= 8
2	×	1	= 2

*continued on next page*

*continued from previous page*

<b>first</b>	<b>×</b>	<b>second</b>	<b>=</b>	<b>RHS</b>
2	×	2	=	4
2	×	3	=	6
2	×	4	=	8
2	×	5	=	10
2	×	6	=	12
2	×	7	=	14
2	×	8	=	16
3	×	1	=	3
3	×	2	=	6
3	×	3	=	9
3	×	4	=	12
3	×	5	=	15
3	×	6	=	18
3	×	7	=	21
3	×	8	=	24

## A.3 Landscape Tables

If your table is especially wide, it may be better to switch it to the landscape orientation. One way of doing this is with the `rotating` package, which implements (among other things) two new environments: `sidewaystable` and `sidewaysfigure`<sup>1</sup>. The way this package achieves this is most useful for *printed results*, as it only rotates the environment on the page (but does not convert the page into landscape orientation)—for electronic viewing of a PDF, it may be useful to rotate the whole page since it's not often easy for the reader to rotate their screen (assuming the sideways content takes

<sup>1</sup>I find `sidewaysfigure` less useful, as it tends to be easy enough to rotate the figure before inclusion, but if the caption/figure are complex it may be useful to have them oriented in the same way

up the whole page). One advantage of this package’s implementation of sideways environments is that it supports `twoside` page layout, and will rotate the sideways environment such that the bottom is towards the outside of the double-page layout in such cases.

An example of a `sidewaystable` is shown in [Table A.2](#)—if you’re reading this as a PDF on your computer, you’ll probably find it difficult to read as it’s sideways on your screen.

## A.4 Including the PDF of a Relevant Paper

Occasionally it may be useful to include whole pages from another document in your thesis, where, for some reason, it is inappropriate or highly inconvenient to convert this into content yourself. This could apply to pages from a technical manual (which would be especially difficult for the average reader to track down), or a highly relevant paper you’ve published in the field, but not exactly on the thesis topic.

Inclusion of a separate PDF at the page level (rather than just as a floating figure) can be achieved using the `pdfpages` package<sup>2</sup>. As an example, (three pages of) “ $P \neq NP$ ”, by Vinay Deolalikar, in its original form are embedded on the following pages.

---

<sup>2</sup>Please note that there appears to be a namespace clash between the `pdfpages` and `graphicx` packages. Including `pdfpages` *after* `graphicx` resolves the issue.

**P  $\neq$  NP**

Vinay Deolalikar  
HP Research Labs, Palo Alto  
vinay.deolalikar@hp.com

August 6, 2010

### Abstract

We demonstrate the separation of the complexity class  $\text{NP}$  from its subclass  $\text{P}$ . Throughout our proof, we observe that the ability to compute a property on structures in polynomial time is intimately related to the statistical notions of conditional independence and sufficient statistics. The presence of conditional independencies manifests in the form of economical parametrizations of the joint distribution of covariates. In order to apply this analysis to the space of solutions of random constraint satisfaction problems, we utilize and expand upon ideas from several fields spanning logic, statistics, graphical models, random ensembles, and statistical physics.

We begin by introducing the requisite framework of graphical models for a set of interacting variables. We focus on the correspondence between Markov and Gibbs properties for directed and undirected models as reflected in the factorization of their joint distribution, and the number of independent parameters required to specify the distribution.

Next, we build the central contribution of this work. We show that there are fundamental conceptual relationships between polynomial time computation, which is completely captured by the logic  $\text{FO}(\text{LFP})$  on some classes of structures, and certain directed Markov properties stated in terms of conditional independence and sufficient statistics. In order to demonstrate these relationships, we view a LFP computation as “factoring through” several stages of first order computations, and then utilize the limitations of first order logic. Specifically, we exploit the limitation that first order logic can only express properties in terms of a bounded number of local neighborhoods of the underlying structure.

Next we introduce ideas from the 1RSB replica symmetry breaking ansatz of statistical physics. We recollect the description of the d1RSB clustered phase for random  $k$ -SAT that arises when the clause density is sufficiently high. In this phase, an arbitrarily large fraction of all variables in cores freeze within



exponentially many clusters in the thermodynamic limit, as the clause density is increased towards the SAT-unSAT threshold for large enough  $k$ . The Hamming distance between a solution that lies in one cluster and that in another is  $O(n)$ .

Next, we encode  $k$ -SAT formulae as structures on which FO(LFP) captures polynomial time. By asking FO(LFP) to extend partial assignments on ensembles of random  $k$ -SAT, we build distributions of solutions. We then construct a dynamic graphical model on a product space that captures all the information flows through the various stages of a LFP computation on ensembles of  $k$ -SAT structures. Distributions computed by LFP must satisfy this model. This model is directed, which allows us to compute factorizations locally and parameterize using Gibbs potentials on cliques. We then use results from ensembles of factor graphs of random  $k$ -SAT to bound the various information flows in this directed graphical model. We parametrize the resulting distributions in a manner that demonstrates that irreducible interactions between covariates — namely, those that may not be factored any further through conditional independencies — cannot grow faster than  $\text{poly}(\log n)$  in the LFP computed distributions. This characterization allows us to analyze the behavior of the entire class of polynomial time algorithms on ensembles simultaneously.

Using the aforementioned limitations of LFP, we demonstrate that a purported polynomial time solution to  $k$ -SAT would result in solution space that is a mixture of distributions each having an exponentially smaller parametrization than is consistent with the highly constrained d1RSB phases of  $k$ -SAT. We show that this would contradict the behavior exhibited by the solution space in the d1RSB phase. This corresponds to the intuitive picture provided by physics about the emergence of extensive (meaning  $O(n)$ ) long-range correlations between variables in this phase and also explains the empirical observation that all known polynomial time algorithms break down in this phase.

Our work shows that every polynomial time algorithm must fail to produce solutions to large enough problem instances of  $k$ -SAT in the d1RSB phase. This shows that polynomial time algorithms are not capable of solving NP-complete problems in their hard phases, and demonstrates the separation of  $\mathbf{P}$  from  $\mathbf{NP}$ .

**Table A.2** – This table is so wide that I decided it should be in the landscape orientation to allow it to fit nicely on one page. You may of course find it easier (for the reader) to reconsider the content and layout of the table, or convert it to a graphical representation, as large walls of data tend to be hard to really interpret well. Almost certainly, you'd only have such large tables in an appendix.

Item	Total	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SA Mission Distance (m)	1101.4	49.4	81.5	34.2	78.8	98.8	70.8	16.0	61.4	14.9	52.1	24.3	83.3	170.3	143.5	20.7	30.1	21.4	99.2
SA Traversed Distance (m)	3244.1	53.9	86.8	90.7	92.1	120.8	74.3	46.4	63.8	15.6	55.3	27.4	127.2	222.4	987.1	273.0	167.0	235.4	505.0
MA Mission Distance (m)	1083.9	59.1	81.5	34.2	78.8	98.8	70.8	16.0	61.4	14.9	52.1	24.3	83.3	170.3	143.5	20.7	30.1	21.4	81.8
MA Traversed Distance (m)	2343.1	61.8	84.6	70.7	84.6	116.6	72.5	46.3	62.6	15.5	53.5	25.8	129.4	213.4	147.1	268.2	174.5	233.8	482.3
Ratio (SA/MA)	1.38	0.872	1.03	1.28	1.09	1.04	1.03	1	1.02	1.01	1.03	1.06	0.983	1.04	6.71	1.02	0.957	1.01	1.05
SA Mission Est. Time (s)	734.3	32.9	54.4	22.8	52.5	65.9	47.2	10.7	40.9	9.9	34.8	16.2	55.5	113.5	95.7	13.8	20.1	14.3	66.2
SA Traversal Time (s)	2436.0	43.5	61.6	70.1	68.0	89.6	55.3	35.2	45.3	12.7	39.8	21.8	93.1	161.8	731.0	204.7	146.1	174.3	382.1
MA Mission Est. Time (s)	1083.9	59.1	81.5	34.2	78.8	98.8	70.8	16.0	61.4	14.9	52.1	24.3	83.3	170.3	143.5	20.7	30.1	21.4	81.8
MA Traversal Time (s)	2411.3	64.2	84.8	73.6	86.8	119.1	74.2	47.2	63.5	16.1	54.3	28.1	132.8	215.2	148.2	271.3	203.4	239.9	488.7
Ratio (SA/MA)	1.01	0.677	0.726	0.953	0.784	0.753	0.746	0.744	0.714	0.786	0.734	0.776	0.701	0.752	4.93	0.755	0.718	0.727	0.782
SA Cost (Exp. Map)	1019924.1	1540.6	49041.0	5094.7	86202.5	98847.3	0.0	7974.8	1773.9	459.0	7825.8	1120.0	4131.3	14618.2	447306.7	19380.5	109612.1	33188.1	131807.7
MA Cost (Exp. Map)	916661.5	6032.4	81939.1	7722.2	73856.9	198949.4	11654.8	6191.5	1398.9	564.8	13336.9	1123.8	6129.8	68857.2	39422.7	3213.2	172319.0	93562.5	130386.5
Ratio (SA/MA)	1.11	0.255	0.599	0.66	1.17	0.497	0	1.29	1.27	0.813	0.587	0.997	0.674	0.212	11.3	6.03	0.636	0.355	1.01
SA Cost (Ground Truth)	26891.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	26891.4	0.0	0.0
MA Cost (Ground Truth)	28400.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	28400.0	0.0	0.0
Ratio (SA/MA)	0.947	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.947	-

Map Configuration	SA Coverage (m <sup>2</sup> )	MA Coverage (m <sup>2</sup> )	Ratio (MA/SA)
Expanded Cost Map	28108	28229	1