



# MCHA3500 project report template

November 2020

Student Name??

<sup>1</sup> *Student of Mechatronics Engineering,  
The University of Newcastle, Callaghan, NSW 2308, AUSTRALIA  
Student Number: 3.....  
E-mail: [First.Last@uon.edu.au](mailto:First.Last@uon.edu.au)*

---

## **Abstract**

Remember that your abstract may include the following information:

- Defines the intention of the report.
- Places the report in context so the reader knows why it is important to read it.
- Why is it important?
- What problem is addressed?
- Briefly states the results
- Briefly presents the implications and recommendations

Ensure that your abstract is less than 200 words.

## **Acknowledgements**

You may like to say thank you to someone that helped you with your project.

## **Contents**

# 1 Introduction

To organise your introduction section you can use the following structure:

- **Position:** Show there is a problem and that it is important to solve it.
- **Problem:** Describe the specifics of the problem you are trying to address
- **Proposal:** Discuss how you are going to address this problem. Use the literature to back-up your approach to the problem, or to highlight that what you are doing has not been done before

Here you need to sell why what you are doing is important, and what benefits will it bring if you are successful and solve the problem?

## 1.1 Subsection title

You can use subsections within any section of the report.

## 1.2 Subsection title

Recall that you need at least two subsections per section.

### 1.2.1 Subsubsection 1

Do not use more than 2 levels of sub-sectioning.

### 1.2.2 Subsubsection 2

Do not use more than 2 levels of sub-sectioning.

The rest of the report is organised as follows. Section ?? describes items related to the core content. Section ?? concludes the report. Appendix ?? shows an example of how to make a Table.

## 2 Core Section

In this section, examples of equations, figures, lists and code are provided.

### 2.1 Mathematics

L<sup>A</sup>T<sub>E</sub>X is very good for writing equations. Equations can be included in-line by using the `$` symbols  $y = mx + h$ . Alternatively, equations can be written separately by using the `equation` environment—see (??) below.

$$y = mx + h. \tag{2.1}$$

When you use the `equation`, the expressions will be automatically numbered. If you do not want a number to be assigned, include an asterisk `*` when beginning the equation environment:

$$z = m_z x^2 + h_z.$$

The `split` command can be used within an equation environment to separate equations over multiple lines. Use an ampersand `&` on each line to specify how the equations should be aligned. Using `split` only provides a single equation number for multiple lines.

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du. \end{aligned} \tag{2.2}$$

If you want to have each line of a multi-line equation numbered, you can make use of the `align` environment,

$$\dot{x} = Ax + Bu, \tag{2.3}$$

$$y = Cx + Du, \tag{2.4}$$

where (??) is the first equation and (??) is the second. If you want to distinguish vectors from scalars you can use **`bold`** for vectors and matrices:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u, \\ y &= \mathbf{C}\mathbf{x} + \mathbf{D}u, \end{aligned}$$

where  $u$  and  $y$  are scalar variables and  $\mathbf{x}$  is a vector variable.

Matrices can be written by using the `bmatrix` command:

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \dots & A_{mn} \end{bmatrix}.$$

Greek letters can be written within equations by using appropriate commands. A few useful examples are:

- $\alpha, \beta$
- $\gamma, \Gamma$
- $\theta, \Theta$
- $\phi, \Phi, \varphi$

You can also write Greek letters in bold using the `boldsymbol` command:  **$\alpha$** .

## 2.2 Figures

Figures can be included in your document by using the `figure` environment. Note that the position of the figure is decided by the latex compiler when creating the final document. Don't be surprised if it does not appear exactly where you expected it. Figure ?? below shows a plot of the function  $\sin(x)/x$ .

Figures can be created using your favourite graphics editor before being included in your LaTeX document. If I need to make a simple diagram, I use powerpoint and select the drawing and save it as a pdf. For example, look at Figure ??.

To import the figures from Matlab, follow the following procedure:

1. Add labels and legends (don't forget to include units in the labels of each axis.)
2. From the file menu tag on the figure select export set up
3. Change the font size to 14 and click apply to figure
4. Export the figure as eps
5. Import it in LaTeX using the include graphics within a `figure` environment.

## 2.3 Lists

To create lists use the environments `itemize`, `enumerate`, or `description`

The following is generated using *itemize*

- This is item 1
- This is item 2

The following is generated using *enumerate*

- 1) This is item 1
  - a) Subitem a
  - b) Subitem b
    - i) Subsubitem i
    - ii) Subsubitem ii

- 2) This is item 2

The following is generated using *description*

**foo)** This is item 1

**bar)** This is item 2

## 2.4 Code listings

To include a syntax-highlighted code listing, you can use the *listings* package. The default options are specified by the `\lstset` command. The default settings for this document have been configured for Matlab code. There are 3 main commands, all of which can include options to override the defaults:

1. `\lstinline`: Command for including code fragments inline with the text, as an alternative to `\verb`. For example, we might describe function prototypes such as `int main(int argc, char *argv[])`.
2. `\begin{lstlisting}...`, `\end{lstlisting}`: Environment for including a source code listing—embedded in the LaTeX source—in a box or floating environment.

```
figure(1)
hold on
grid on
% Plot the input voltage
plot(time,voltage)
% Plot the recovered voltage
plot(time,Ra*current + Kw*velocity,'+')
```

The default style can be changed using the `style` command. A `CStyle` option has been defined within this template for `.c` code.

```
1  int add_function(int x, int y)
2  {
3      /* Add inputs and return value */
4      return x+y;
5  }
6
```

3. `\lstinputlisting`: Command for including a source code listing—loaded from an external file—in a box or floating environment. An example is shown in Listing ??.

### 3 References and Citations

Bibliographies are generated by LaTeX when the `bibliography` command is used at the end of the document—see the bottom of the template. The `bibliography` command requires you to provide a `.bib` file that contains all of the reference used in the document. The format of the bibliography and citations can be specified by using the `bibliographystyle` command.

Bibliography `.bib` files can be created with a reference manager such as [JabRef](#). To include a reference in your document, use the `cite` command with the appropriate reference key. The appropriate key can be found within your reference manager. Alternatively, most reference editors have a short-cut to copy the cite command with key directly. In JarbRef, select the reference you want and press 'ctrl+k' to copy the cite command with the corresponding key.

Variations on the `cite` command are `\citet{}` and `\citep{}`—?, (?), ?. You can combine more than one reference in a single citation (??).



## 4 Conclusion

This is one of the most important parts of the report. In the conclusion section, you should

- briefly summarise the results,
- reflect on the work presented,
- make recommendations,
- suggest future work or improvements.

## A Example of a Table

Table 1: Bachelor of Engineering Mechatronics First Year Program

<b>1st Year</b>		
Semester	Course Code	Course Name
1	GENG1003	Introduction to Procedural Programming
1	GENG1500	Introduction to Professional Engineering
1	MATH1110	Maths for Engineering, Science & Technology 1
1	PHYS1205	Advanced Physics I
2	CIVL1100	Fundamentals of Engineering Mechanics
2	ELEC1310	Introduction to Electrical Engineering
2	MATH1120	Maths for Engineering, Science & Technology 2
2	MECH1110	Mechanical Drawing/CAD & Workshop Practice