Skripsie guide

J.C. Schoeman

October 24, 2023

Contents

Chapter 1

Introduction

This document is written for my skripsie students, but can be distributed to anyone else as desired. I expect that all my students read through Chapters 1 and 2 at the start of their project and through Chapters 3 and 4 before they start writing their report, with multiple rereads as necessary. This guide serves two main purposes:

- To explain various aspects of a skripsie such as the timeline and report.
- To provide a guide to (and example of) proper technical writing using LaTeX.

It is, however, not a template for the skripsie report itself. Such a template can be found on the course's webpage. Also make sure to read the module framework thoroughly and refer to the webpage for any practical information such as ordering components and printing PCBs.

In general, a skripsie is the student's own responsibility. The role of the supervisor is mostly to propose a suitable topic and to provide guidance along the way. The student will then complete the project themselves and must make sure that they satisfy all the required ECSA Graduate Attributes. An anonymous colleague once said: "Think of your skripsie supervisor as a bad manager at your first job. They (a) present a problem which you have to solve, (b) only check in occasionally and (c) expect a good result after a short deadline."

In my opinion, there are five things that you need to show to complete a skripsie successfully:

1. That you can identify a challenging and relevant engineering problem.
2. That you can do independent research on possible solutions and additional concepts not covered during undergraduate studies.
3. That you can apply your engineering and scientific knowledge to design an appropriate solution.
4. That you can implement the solution, evaluate it using appropriate experiments and improve upon it if necessary.
5. That you can communicate professionally – both verbally and in writing.

At the end of the day, a skripsie must also be enjoyable. This is the culmination of your undergraduate studies and your chance to show the world what you have learnt. Therefore, start early, work hard and you will never regret it.

Chapter 2

Project overview

This chapter explains various practical arrangements for a skripsie. Most of these aspects are according to my own preference and I am happy to receive feedback on where things can be improved. After discussing weekly meetings, I provide a summary of the skripsie timeline and then discuss each phase in more detail.

2.1 Weekly meetings

By default, I schedule a 30-minute, weekly meeting with each of my skripsie students. These meetings will start in the first week of the second semester. The purpose of such a meeting is threefold:

- To receive an update of what the student did in the past week.
- To answer any technical or general questions that the student may have.
- To check that the student has a plan for the coming week.

Depending on the stage of the project, however, the focus and length of the meetings may vary. For example, at the start of the semester the meeting could be longer to explain the problem statement as well as new technical concepts. In contrast, meetings are typically shorter towards the end and focus mainly on the report and supervisor feedback.

Once per month, I replace the default individual meetings with a longer group sessions to discuss aspects that are common to all skripsies, for example report writing. I also ask each student to give an update on their progress, as well as what they are currently struggling with, to the whole group. This provides some experience in communicating your work to external people, as well as an opportunity to get additional input or answers to your questions.

2.2 Timeline

Typically, an E&E skripsie is about four months long, form middle July to middle November. It is, once again, the student's own responsibility to make sure they

meet all the deadlines such as the student-supervisor agreement and the final hand-in. Although you should receive your topic by June, this is only a second-semester module. Therefore, first focus on the first-semester exams. If you want to start working in the July holiday, that is your choice. Of course, the earlier you start, the more you can achieve.

A typical timeline of a skripsie is shown in Table 2.1. This is only a rough overview of the phases and you have to set up your own detailed plan specific to your project. Also note that different phases often overlap. For example, you typically start writing the report while still performing experiments on the side. The most important deadline is the final hand-in of the report and one usually works back from this to make sure everything occurs on time. If you require components, you also need to order this as soon as possible to avoid delays.

Month	**Weeks**	**Task**	**Description**
July	2	Literature review	Identify the problem and investigate existing solutions
August	4	Design solution	Learn the necessary theory, design the system and order components
September	4	Implement solution	Build and test the subsystems, combine everything and improve
October	4	Write report	Write the first draft using LaTeX and incorporate supervisor's feedback
November	2	Examination	Submit report, prepare presentation and poster and present these

Table 2.1: Typical timeline for a skripsie.

2.3 Literature review

The first phase of your project is a thorough literature review. This refers to doing your own research to (a) better understand your problem statement and to (b) investigate existing solutions to this problem. There are three main reasons for the latter: Firstly, you will need to compare your eventual solution to existing solutions based on appropriate criteria. Secondly, one of the best ways to design a solution is to build on what others have done. Lastly, this is one of the elements of a good skripsie as outlined in Chapter 1.

When it comes to sources that you can use for your research, it is best to use articles and textbooks. Only use a website in your bibliography if you have no other option. That being said, Wikipedia is often a good place to start reading about a new concept, field or technique. Once you know the basic terminology, you can then perform better keyword searches to find more credible sources. One of the best tools for such searches is Google Scholar.

At this stage of your project, it is not yet necessary to write a complete literature review (Chapter 2 in your report). If you do this too early, it is still difficult to know

which direction your project will take and therefore you run the risk of writing about unrelated literature. If you start too late, however, much of this research will no longer be fresh in your memory by that time. A good idea is to make rough notes about every article you read that you think could be relevant. This could be a pargagraph, bullet points or even keywords that you can use as a starting point later on. This is a good time to start learning LaTeX as well. Another useful software tool for managing research and references is Mendeley.

2.4 Design and implementation

The majority of your time will be spent on designing, implementing and testing your solution. This is often referred to as the "work" component of your project. However, the exact details of this phase will differ from project to project. If you have a purely software-based skripsie, this will involve setting up simulation environments, algorithm design, running computer experiments and often contains a larger theoretical component. If your scope includes hardware, you will typically write less code, but need to design circuits, order components, assemble the system and run practical tests.

2.5 Report writing

The most important output of your entire skripsie is the report. Unlike with E-design, you will not have a final demonstration of your system. However, your supervisor will still want to see a working system before you hand in. You need to start with your report well before the deadline and give it the attention it deserves. If you do an excellent project, but the report is not up to standard, this will affect your final mark severely.

The typical structure of the report is discussed in Chapter 3 and various components of good technical writing in Chapter 4. Although the report can be written in Microsoft Word as well, I strongly recommend that you use LaTeX. The main reasons for this are (a) that the formatting is separated from the content, (b) referencing is automatic, (c) mathematical equations are much easier to write and copy and (d) there are good templates available. The finished product also looks more professional with LaTeX. Although LaTeX has an initial learning curve, I can guarantee that it will save you valuable time towards the end of your project. Just imagine not having to check and edit all your references, page numbering and figure layouts in MS Word after you add one small figure on page 1.

2.6 Feedback

Unlike previous technical reports that you have submitted for other modules, you must submit a draft version of your skripsie report to your supervisor before the final hand-in. Each supervisor handles this in a different way, but I prefer to receive this draft in stages: First, Chapters 1 to 3 (four weeks before hand-in), then Chapters 4 to $n-2$ (three weeks before hand-in), and finally the last two chapters plus the

abstract (two weeks before hand-in). This way, you can work on the next chapters while I review the previous ones. This also provides intermediate deadlines to make sure you are on track to finish in time.

When sending chapters for feedback, send the entire document in PDF format and tell me which chapters to look at. I will use a PDF editor (such as Okular) to make annotations and will send it back as soon as possible (usually within two weeks). The purpose of the feedback is to help you improve the report based on my impressions. You should only send me complete chapters that you are 100% happy with and which have been thoroughly proofread. I will only look at each chapter once, so use this opportunity wisely.

The supervisor feedback is technically only a suggestion. It is ultimately still your report, but since your supervisor is also one of your examiners, it is a good idea to incorporate the feedback as far as possible. The student must also take full responsibility for the final version of the report. Not receiving feedback on a particular section, sentence or figure does not necessarily mean it is perfect. I often do not mark the same mistake (e.g. capitalisation, abbreviations, etc.) more than once. You should therefore fix all similar occurrences before sending later chapters for feedback.

2.7 Examination

For the exact details and arrangements of the skripsie examination process, refer to the module framework and announcements on the webpage. In general, there are two examiners for each skripsie student. One is your supervisor and the other is an internal examiner from the E&E department. Each examiner will read your report independently and propose a preliminary mark. Within a week or two after the report hand-in, there will then be an oral examination. First, the student will give a 10-minute presentation on their project, which is open for the public to attend. Afterwards, the examiners can ask questions before deciding on a final mark together. Once the skripsies have been externally moderated at the open day, for which you need to prepare an A1 poster about your project, the marks will be loaded onto the normal system.

Chapter 3

Report structure

This chapter outlines the report structure that I personally prefer. According to the specific project and the student's own preferences, variations of this are also possible. Also make sure you look at good examples of previous skripsie reports for inspiration. Writing the report is inevitably an iterative process: First, you plan a rough outline. Next, you start writing the chapters (preferably in chronological order). After each section or chapter, you should then revisit your report outline, rinse and repeat. The more you have on paper, the easier it is to organise the ideas in a logical order.

Always remember that the target audience of your report is someone with a BEng degree in E&E Engineering. You therefore need to explain any additional theory or concepts not covered during undergraduate studies clearly, but avoid covering standard knowledge in too much detail. Report writing is all about balance. You need to use the 40 pages as best as possible to make sure you meet the five criteria outlined in Chapter 1.

3.1 Introduction

In the first chapter of your report, you should have at least four sections. These are the *1.1 Background and motivation*, *1.2 Problem statement and objectives*, *1.3 Solution overview and scope* and *1.4 Report outline*.

In the first section, you need to explain the bigger picture. The question to answer here is: "Why does your project matter?". This section should introduce all the high-level concepts that is necessary to understand your problem statement. Do not focus on your chosen solution here yet. In the next section, you will then use this context to define the aim of your project and to identify the concrete objectives. These objectives are measurable sub-goals that will help you to solve this problem.

In the third section, you should provide a brief overview of your chosen solution and scope, without going into too much detail or discussing the final results. This helps the reader to know what to expect from the beginning. Make it clear that the motivation for this solution is still coming later. Finally, you should add one or two paragraphs that outline what is discussed in each future chapter and how it fits into the overall story of the report. Write two or three sentences for each chapter, and make sure there is a logical link running through this section.

3.2 Related literature

The purpose of the second chapter is to answer the question: "What have others done to solve this or similar problems?". In this chapter, you should provide high-level overviews of multiple sources rather than in-depth discussions of a few. The closer an existing solution is to your problem statement and application, the more attention it should get. A good idea is to divide the sources into three or four categories and use these as the section headings. For each approach, you should summarise which problem the authors solved, what they did, how well their solution performed and what its shortcomings are with regards to your own problem statement.

Common mistakes for this chapter are to go into too much unnecessary technical detail about how the existing solutions worked, to only list the performance of each without discussing or comparing it, or to discuss each solution in isolation with no logical arguments building on one another. A final subsection titled *2.n Evaluation of existing approaches* works well to highlight the most relevant solutions and to motivate why you decided on your solution which you will introduce over the next few chapters. Only include mathematical equations and technical detail in this chapter if it is necessary for this evaluation or motivation.

3.3 System overview

This third chapter provides the detail on how you modelled the problem, as well as the assumptions and high-level choices that you made. A *3.1 System diagram* section provides a convenient way to identify and discuss the different subsystems of your solution as well as the interfaces between them. You should also identify which subsystems do not have to be designed (if any) in this section, but do not discuss the details of the subsystems yet. You are effectively formalising the sub-problems which you needed to solve at this point. Next, you need to mention the *3.2 Requirements and metrics* that you will use to evaluate the solution.

For a more software-intensive project, an important aspect of this chapter could be the setup of the *3.3 Simulation environment*, although this should not go into the detail of your implementation. Finally, for each subsystem (or groups thereof) that you designed, you need to motivate why you chose the particular high-level solution. This works well in a dedicated section for each subsystem (or group). It is a good idea to compare multiple options for each and to refer back to the *2.n Evaluation of existing approaches* section from Chapter 2.

3.4 Chapter for each subsystem

The number of these "middle" chapters will vary depending on the project. Once you have described the complete system and introduced each subsystem at a high level in Chapter 3, you need to provide the details of your solution. In a hardware-focussed project, this includes various circuit diagrams and calculations of component values. For a more software-focussed project, the focus will be on architecture design, flow diagrams and hyper-parameter values. By the end of these chapters, the reader

should be able to implement your solution to duplicate your results. Make sure that your design choices are properly motivated in this chapter.

What works well is to divide your solution into two or three logical parts, where each then becomes a chapter. This division may be difficult to do from the start, so first write it as one chapter and then try different combinations. In each chapter, it is important to clearly separate the work that is your own from the general theory that is not, which should be collected at the start of the chapter in a dedicated section or two. Such sections are typically more mathematical, but should include references, since this is not your own work. Think of these as summaries of the textbook(s) that you used, but that only contains the absolutely necessary information. A good rule of thumb is that an equation might not be appropriate to include if you never have a cross reference to it later on.

3.5 Experiments and results

To show that your project was successful, you need to perform suitable experiments, present the results in an understandable way and provide meaningful discussions thereof. This chapter should be linked to the requirements and metrics mentioned in Chapter 3. You should also refer to the literature in Chapter 2 to support the metrics you used and to compare your results to that of existing approaches. Avoid simply stating that "things worked", but instead think critically about any aspect someone might wonder about (e.g. power usage, execution time, robustness, statistics or scalability). Include both qualitative (e.g. photos) and quantitative results (e.g. graphs and tables).

3.6 Conclusion

In the final chapter, you should give a summary of the problem statement, your solution and the results. However, be careful not to sound too repetitive. In essence, this is where you tie everything together to show how the results from the previous chapter satisfy the objectives in Chapter 1. You should evaluate your solution critically and focus on its implications. Finally, conclude by discussing future work that could build on this project. This chapter should answer the questions: "So what?" and "What now?".

Chapter 4

Technical writing

This chapter covers useful guidelines to proper technical writing. When reading a skripsie report, an examiner's first impressions are very important. You therefore need to make sure that your report not only contains all the technical details, but also looks professional and reads easily.

4.1 Structure and layout

Although your report contains multiple chapters, sections and subsections, there should be one main story from start to finish. Your arguments should therefore build on one another in a logical order and the transitions between sections and chapters should be smooth. Headings are there to add structure and help someone to navigate quickly between sections, but do not rely on these alone to give the context for a certain section.

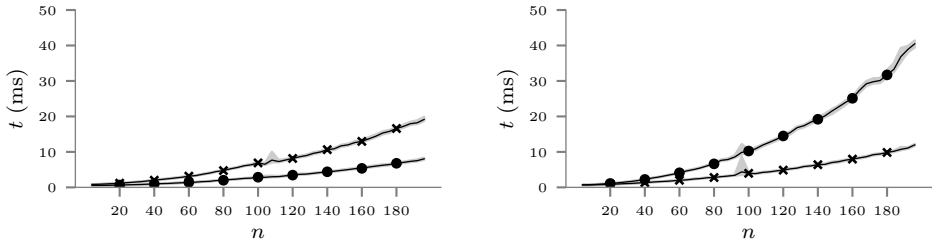
At the start of each chapter, before the first section heading, it is recommended to add an introductory paragraph. The purpose of this is (a) to give an overview of what will be presented in this chapter and (b) to refer back to the introduction and previous chapters to show where this chapter fits into the overall story. Similarly, use one or two introductory sentences at the start of each section.

It is very important not to sound repetitive. Therefore, do not use similar headings throughout the report, do not start each chapter the same way and do not repeat the same phrases over and over. This is especially important at the end of a chapter, where you need to zoom out from the technical detail, summarise the most important points from this chapter and link back to your overall story. I do not particularly like a dedicated section at the end of each chapter called *Summary*. Instead, simply write one or two paragraphs at the end of the last section that serve these functions.

The body of a skripsie report mainly consists of three elements: figures, equations and normal text. You should carefully consider how to use these in the most appropriate way. For example, a figure is the fastest way to convey information, but is not very precise. Mathematical equations, on the other hand, are very precise, but could take longer to understand. Normal text (i.e., paragraphs) is somewhere in between and is used most commonly in technical writing. However, avoid too many paragraphs in a row without any headings, figures or equations to refer to.

4.2 Figures

All figures should be numbered (LaTeX does this for you automatically), have a descriptive caption and be referred to at least once in the text *before* the figure appears. Figure 4.1 shows an example of this. The figure caption can be multiple sentences and does not have to be a single line. The purpose of the caption is to say exactly what is shown in the figure – including (but not limited to) what certain colours, symbols or lines refer to, what appears on each axis and what the most important things are to take note of. In the case of subfigures, assign a letter to each and then refer to this in the overall caption.



(a) (b)

Figure 4.1: Measured execution times (in milliseconds) of (a) marginalisation and (b) multiplication using our degenerate Gaussian factors (indicated by the lines with x-shaped markers) and those proposed by Raphael (indicated by the lines with circular markers) as a function of the input dimension n. The 25'th and 75'th percentile (based on multiple experiments) are indicated by the shaded regions.

A figure needs to be understandable just by reading its caption, without having to read the main text. This is useful for someone who wants to briefly scan through your report (and especially your results) and decide whether they should read it in detail. Alternatively, someone might want to page back to a section they already read in detail and quickly look at certain figures again. Avoid repeating anything you say in the figure caption in the main text as well. When in doubt, rather include it in the caption. It is often appropriate to refer to a figure more than once from the main text.

4.3 Tables

Tables are used to represent different types of data than figures. This includes qualitative attributes (such as advantages or disadvantages) and summaries of results (such as values of certain hyperparameters). Usually, if you can choose between a figure and a table, opt for the figure. Figures are better at representing trends or comparing a lot of numerical data simultaneously. The same rules as for figures apply to tables when it comes to captions, referencing, etc., as shown in Table 4.1.

Operation	Input	Output	Complexity
Sum	$m \times n$ and $m \times n$	$m \times n$	$\mathcal{O}(mn)$
Product	$m \times n$ and $n \times p$	$m \times p$	$\mathcal{O}(mnp)$
Inverse	$n \times n$	$n \times n$	$\mathcal{O}(n^3)$
Pseudo-inverse	$m \times n$, $m \geq n$	$n \times m$	$\mathcal{O}(mn^2)$
Determinant	$n \times n$	scalar	$\mathcal{O}(n^3)$
SVD	$m \times n$	$m \times m$, $m \times n$ and $n \times n$	$\mathcal{O}(\max[m^2n, mn^2])$
Column space	$m \times n$, rank r	$m \times r$	$\mathcal{O}(\min[m^2n, mn^2])$
Nullspace	$m \times n$, rank r	$n \times (n-r)$	$\mathcal{O}(mn^2)$
Complement	$m \times n$, $\perp$	$m \times (m-n)$	$\mathcal{O}(m^2n)$

Table 4.1: Asymptotic time complexity of common matrix operations in terms of the input and output dimensions of the applicable matrices.

4.4 Equations

There are two options when including an equation (or any mathematical expression or symbol) in your text: either in-line or on its own line. Generally, I prefer the former when the equation is short, you need to save space and it is not necessary to refer to it later on. In this case, use "$...$" to yield for example $ax + b = c$. For anything longer than this, use the `equation` or `align` command which will place the equation on its own line and assign a number to it, which you can then refer to.

Before (or immediately after) you use any symbol in an equation, you need to say what it represents. The easiest way to do this is to include the symbol into a sentence where you mention the corresponding variable. You should also make equations part of full sentences and not refer to standalone equations as you would for figures or tables. The reason is that an equation does not have a caption, and therefore is difficult to understand if viewed on its own. To know where to place punctuation marks in equations, read the symbols aloud as part of the sentence. Here is an example:

"For a univariate scalar function $g(x)$, recall the Taylor series

$$g(x) = g(a) + \frac{g'(a)}{1!}(x-a) + \frac{g''(a)}{2!}(x-a)^2 + \frac{g'''(a)}{3!}(x-a)^3 + \dots \tag{4.1}$$

around a given point a. Such an expansion can be truncated to obtain a first-order approximation of a nonlinear function around the mean vector $\boldsymbol{\mu}$, namely

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\boldsymbol{\mu}) + J(\boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu}), \tag{4.2}$$

where $J(\mathbf{x})$ is the Jacobian matrix of the function $\mathbf{f}(\mathbf{x})$."

4.5 Referencing

At the end of your skripsie report you should have a bibliography that contains the references you used throughout the report. There are typically three use cases

for a reference: Firstly, to describe and credit existing solutions to your problem. Secondly, to support claims that you make without motivation or proof. Thirdly, to provide an additional source for someone who wants to read further on a particular topic. Chapter 2 will typically have the most references, followed by Chapters 1 and 3.

I prefer the IEEE reference format, where the bibliography is organised according to numbers in square brackets. The order of the references should be the order in which they appear in the report. When using a reference in the text, do not use this number as a noun. In other words, the sentence should be a full sentence even if you omit the reference number. Here are two examples: "An important (and necessary) constraint for using Gaussian models is that the covariance matrix of any Gaussian distribution must be positive definite [1]." and "Two examples include the work by Pawula et al. [2], where they determine formulas for the symbol error rate in digital frequency modulation, and that by Cao and Shen [3] on the fault detection of photovoltaic cells using Gaussian mixture models (GMMs)."

4.6 Algorithms

An algorithm is a good way to group many mathematical equations together. This can be very useful when you discuss a standard technique that you used or as a summary at the end of your own derivations. What works especially well is to introduce the algorithm in the main text (possibly with a reference) and briefly say what it does. After the algorithm, you then explain the details line by line.

4.7 Software

Although software development (i.e., coding) could be a significant part of the work that you do for a skripsie, your report should contain relatively few code snippets or listings (if any). If the codebase is an implementation of mathematical operations, rather write about the mathematics. Use pseudocode and flow diagrams instead of inserting entire code listings and do not refer to the programmatic names of variables, functions and files in the main text. You can include more detailed documentation of your codebase in the appendix if you want, but remember that reading an appendix is always optional for any reader.

4.8 Abbreviations

When using abbreviations for the first time, you need to write out what the abbreviation stands for in full, followed by the abbreviation in brackets. After this, you can then use the abbreviation instead of the full version. The full version should be capitalised according to normal rules – both in the main text as well as in the nomenclature. Also use a lower-case 's' to make any abbreviation plural. Here is an example: "To keep the context general enough, we make use of probabilistic graphical models (PGMs) for this purpose. PGMs are a family of statistical techniques that represent the factorisation of a problem using directed or undirected graphs."

Bibliography

[1] P. Z. Peebles, *Probability, random variables, and random signal principles*. McGraw Hill, 1987.

[2] R. Pawula, S. Rice, and J. Roberts, "Distribution of the phase angle between two vectors perturbed by Gaussian noise," *IEEE Transactions on Communications*, vol. 30, no. 8, pp. 1828–1841, 1982.

[3] Q. Cao, C. Shen, and M. Jia, "A fault detection scheme for PV panels in large scale PV stations with complex installation conditions," *arXiv preprint arXiv:2105.08943*, 2021.