

UFFEMA DEVELOPERS

UFFEMA:

Unified Framework for Electric Machine Analysis

REFERENCE MANUAL VERSION 0.1

Dr. A. J. Piña Ortega

October 14, 2018

Copyright © 2018 Dr. A. J. Piña Ortega

POWERED BY UFFEMA DEVELOPERS

<https://github.com/ajpina/uffema>

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

First printing, October 2018

Contents

1	<i>Installation</i>	11
2	<i>Rotating Machines</i>	13
3	<i>Slots</i>	15
4	<i>Windings</i>	17
5	<i>Permanent Magnet Synchronous Machines</i>	19
6	<i>Material</i>	23
	<i>Bibliography</i>	25
	<i>Index</i>	27

List of Figures

1	Composition of rotating electric machines	9
2	Classification of rotating electric machines	10
2.1	Stator architecture.	13
3.1	Parameters for slot type 0.	15
3.2	Type 0. Points, lines and loops.	16
3.3	Parameters for slot type 1.	16
3.4	Type 1. Points, lines and loops.	16
5.1	This graph shows $y = \sin x$ from about $x = [-10, 10]$. Notice that this figure takes up the full page width.	19
5.2	This is a margin figure. The helix is defined by $x = \cos(2\pi z)$, $y = \sin(2\pi z)$, and $z = [0, 2.7]$. The figure was drawn using Asymptote (http://asymptote.sf.net/).	20

List of Tables

5.1	Table caption text	21
-----	--------------------	----

Introduction

The **Unified Framework For Electric Machine Analysis (UFFEMA)** describes with parameters the geometries found in electric machines. Although it would be unrealistic cover all possible shapes or all small details that creative designers put into their models when trying to enhance specific machine behaviours, this document will try to gather the most common geometries of the most common rotating electric machines.

This reference should become a living document that will be incorporating features and evolving such as electric machines have done it since their conception. The present manual corresponds to UFFEMA software version 0.1.

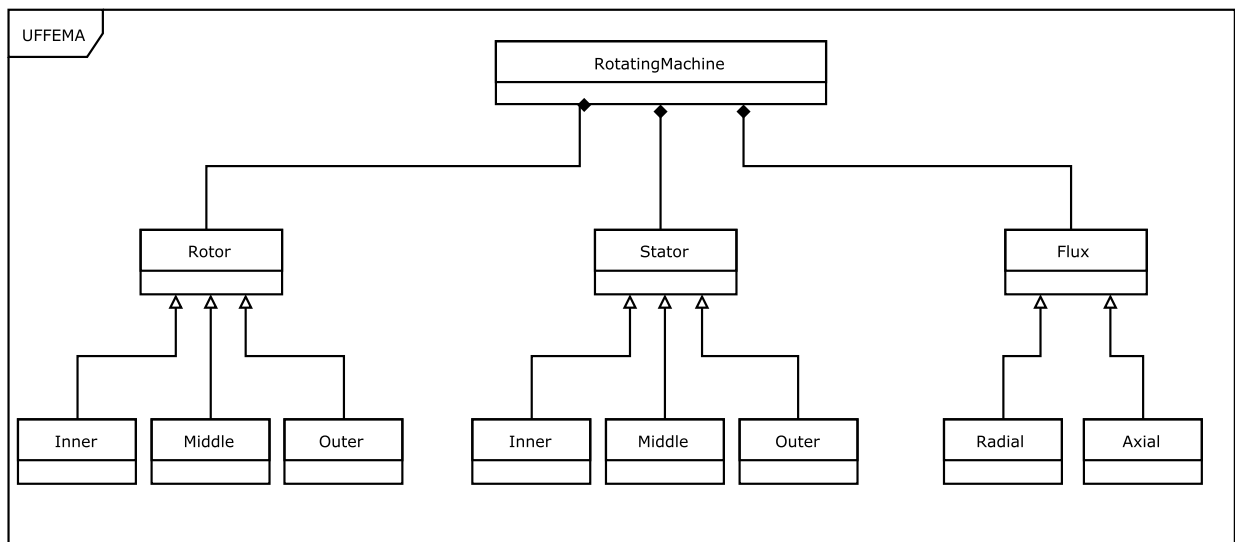


Figure 1: Composition of rotating electric machines

Throughout this document, rotating electric machines will be referred simply as electric machines, so the perspicacious readers will probably complain on that terminology since linear machines or transformers are not treated here. Hence, we apologise for appropriating the definition.

The abstraction of Figure 1 shows the first level of simplicity of electric machines and the sort of relationships will be found along this manual. From this definition, a *RotatingMachine* must have one or more *Rotor*, *Stator*

as well as *Flux*. Either *Rotor* or *Stator* might be of *Inner*, *Middle* or *Outer* construction.

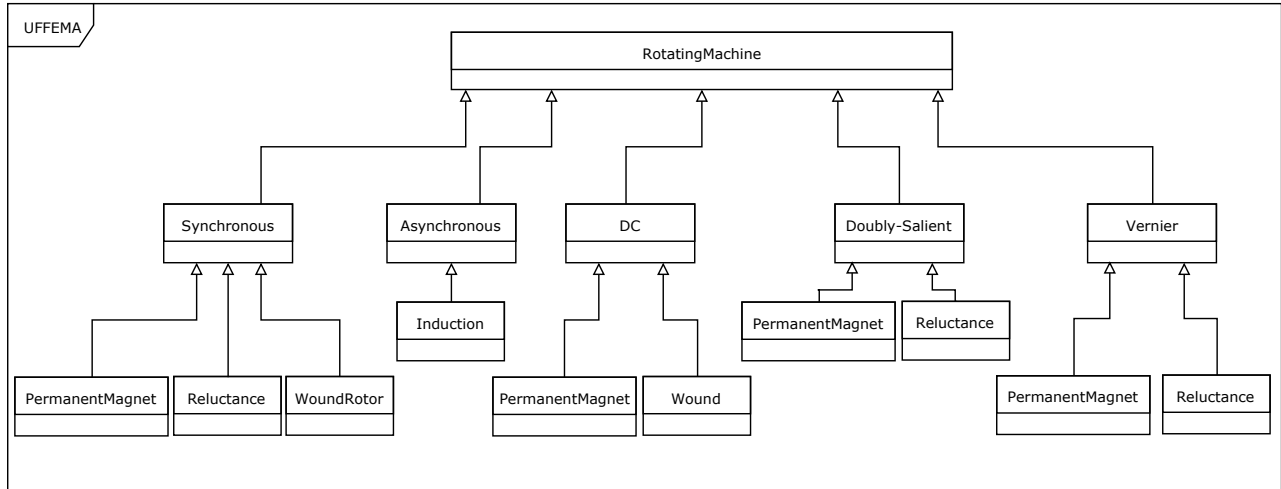


Figure 2: Classification of rotating electric machines

Initially, this document will be dealing with geometries that can be included on machines of *Radial* or *Axial Flux*.

UFFEMA follows the oriented-object paradigm in order to provide the users and developers with a set of objects that can be used to fully define an electric machine. These objects can be serialized and sent to solvers based upon either analytical or numerical methods.

UFFEMA is a framework independent of solving method, but intended to be seamlessly integrated with open-source solvers. At the moment of writing this manual, UFFEMA is being developed in parallel with EMANPY¹ and EMANFES².

At first, a classification for electric machines needed to be agreed in order to be followed and develop the framework. Therefore, rather than trying to make up a new one, it was better idea to review some books on history and new trends on rotating machines.

A classification for electric machines has been given by ³, which includes not only conventional machines such as induction and permanent magnet but it also adds recent developments on flux-switching and vernier machines.

An excerpt of the classification is shown in Figure 5.1, each chapter of this manual deals in further detail with a type of electric machine, examples are also given for each machine.

¹ A.J. Pina Ortega. *EMANPY: Electric Machine Analysis with Python, Reference Manual*. 2018b

² A.J. Pina Ortega. *EMANFES: Electric Machine Analysis with Finite Element Solvers, Reference Manual*. 2018a

³ K.T. Chau. *Electric Vehicle Machines and Drives: Design, Analysis and Application*. Wiley, first edition, 2015. ISBN 978-1-118-75252-4

1

Installation

1.1 How to Get UFFEMA

The source code of UFFEMA is available for free download from its repository in GitHub at <https://github.com/ajpina/uffema> and is licensed under the Apache License, Version 2.0. Repository can be cloned as follows,

```
$ git clone https://github.com/ajpina/uffema
```

1.2 Requirements

A rotating machine is set through a configuration file. The framework is being developed with Python 3.6 and configuration files must be in JSON format. The following dependences need to be met in order to run UFFEMA:

- Python 3.6
- Numpy 1.13

1.3 Running UFFEMA

Being a framework, UFFEMA does not have graphical interface and the only output that produces is a python object that contains information regarding geometry, materials and connection of a rotating electric machine. As such, the objects are intended to be embedded in your own python code. However, a command-line-based program is included to test the installation. The program reads an electric machine definition file in JSON format and prints a string version of the python object created. The input files are usually given with extension `'msf'` (machine settings file) even though there is no need to specify the extension, it helps to keep the machines organized in the working folder.

```
$ uffema-test options [filename]
```

The following options are currently supported by **uffema-test**:

-v

Prints the version.

-f

filename

Reads the machine settings file and creates a python instance.

2

Rotating Machines

A rotating machine is set through a configuration file. The framework is being developed with Python 3.6 and configuration files must be in JSON format. In general, an electric machine consists of a stator and a rotor, this section of the manual deals with the stator's configuration, which is very similar in conventional machines. A separated chapter will be added in order to address stators of such machines that does not fit into this category, for instance, recent developments have aimed to insert permanent magnets or field windings into the stator.

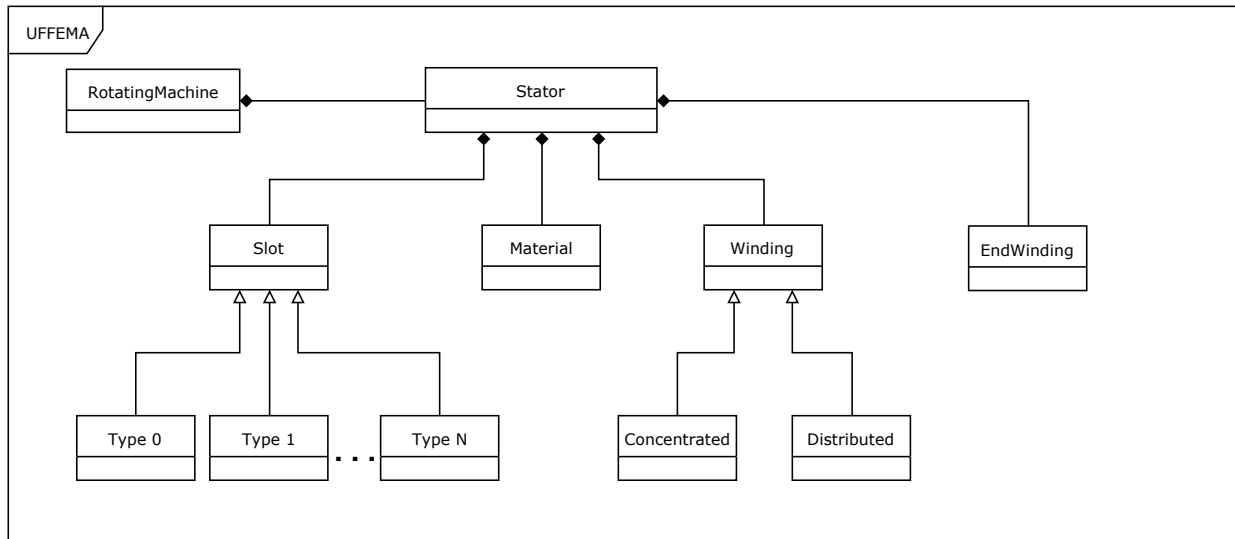


Figure 2.1: Stator architecture.

Several chapters are included in this manual in order to explain each type of machine technology. Moreover, the rotor architecture is let to be explained in those chapters since its geometry is mostly defined by the type of rotating machine.

2.1 Stator

As can be seen in Figure 2.1, at least one stator is found in a rotating machine and is characterized by having slots, windings in its slots, as well as made of a material (usually soft magnetic). The parameters recognized by UFFEMA are as follows,

type: <i>String</i>	Type of stator, for instance, <i>standardouter</i> , <i>standardinner</i> , etc.
oSr: <i>Float</i>	Stator outer radius.
iSr: <i>Float</i>	Stator inner radius.
Ns: <i>Integer</i>	Number of slots.
Sl: <i>Float</i>	Length of stator stack.
slots: <i>Object</i>	Settings for stator slots according to object detailed in chapter 3.
material: <i>Object</i>	Settings for stator material according to object detailed in chapter 6.
winding: <i>Object</i>	Settings for stator windings according to object detailed in chapter 4.

The example for the settings file in JSON format is given below,

```

1 {
2   "machine" : {
3     "type" : String,
4     "stator" : {
5       "type" : String,
6       "oSr" : Float,
7       "iSr" : Float,
8       "Ns" : Integer,
9       "Sl" : Float,
10      "slots" : {...},
11      "material" : {...},
12      "winding" : {...}
13    },
14  }

```

3

Slots

The number of slots where the conductors are to be inserted are defined in the parent component, either stator or rotor. However, the "slots" object must contain type and dimension. Additional parameters for the position of slots on the parent component, for instance, **S0pos** and **Spos** are inserted in the **dimension** field. These are angles for the placement of slot openings and coil regions, respectively. Even though these might be calculated most of the cases with the total number of slots, are particularly useful for an asymmetrical arrangement.

type: *String* Type of slot, for instance, *type0*, *type1*, etc.

dimension: *Array* List of parameters that define the geometry according with its **type**.

```
1 "slots" : {  
2   "type" : String,  
3   "dimension": [  
4     {  
5       "S0pos" : Float,  
6       "Spos" : Float,  
7       ...  
8     },  
9     {  
10      "S0pos" : Float,  
11      "Spos" : Float,  
12      ...  
13    }  
14  ]  
15 }
```

Other parameters follow the two aforementioned, these depend on the type of the slot selected. More detail about the parameters for each slot supported by UFFEMA are described in the next sections.

3.1 Type 0

The following parameters set forth slot opening, wedge and coil regions for slots type 0 (type0), according to Figure 3.1 and 3.2, are also

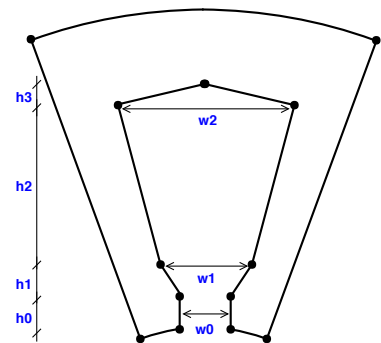


Figure 3.1: Parameters for slot type 0.

embedded in **dimension** field. The latter figure shows points, lines and loops in order to build surfaces if the geometry is to be meshed.

w0 :	<i>Float</i>	Slot opening width.
w1 :	<i>Float</i>	Wedge region width.
w2 :	<i>Float</i>	Coil region width.
h0 :	<i>Float</i>	Slot opening height.
h1 :	<i>Float</i>	Wedge region height.
h2 :	<i>Float</i>	Coil region height.
h3 :	<i>Float</i>	Slot bottom height.

3.2 Type 1

The following parameters set forth slot opening and coil regions for slots type 1 (type1), according to Figure 3.3 and 3.4. Note that slots of type 1 do not show wedge regions and toothtips are concentric. These parameters are also embedded in **dimension** field. The latter figure shows points, lines and loops in order to build surfaces if the geometry is to be meshed.

w0 :	<i>Float</i>	Slot opening width.
w1 :	<i>Float</i>	Wedge region width.
w2 :	<i>Float</i>	Coil region width.
h0 :	<i>Float</i>	Slot opening height.
h2 :	<i>Float</i>	Coil region height.

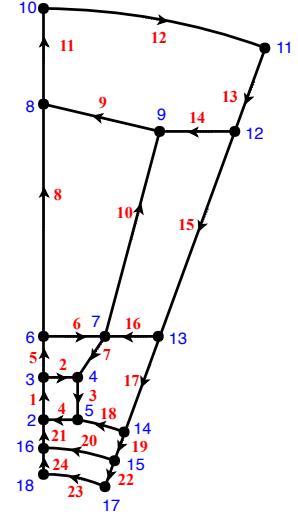


Figure 3.2: Type 0. Points, lines and loops.

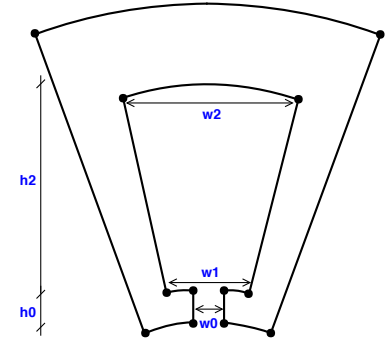


Figure 3.3: Parameters for slot type 1.

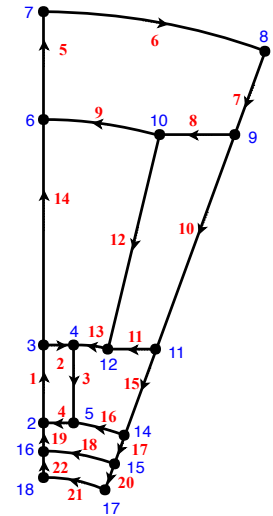


Figure 3.4: Type 1. Points, lines and loops.

4

Windings

EXAMPLE OF the newthought command for starting new sections. Typography examples: ALL CAPS and SMALL CAPS.

5

Permanent Magnet Synchronous Machines

A rotating machine is set through a configuration file. The framework is being developed with Python 3.6 and configuration files must be in JSON format.

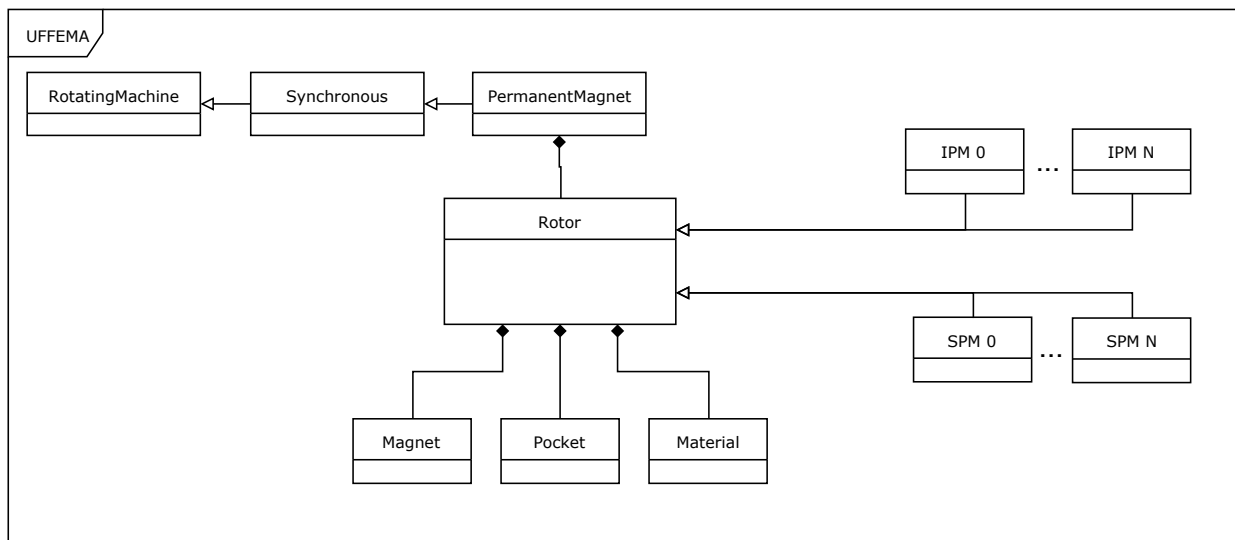


Figure 5.1: This graph shows $y = \sin x$ from about $x = [-10, 10]$. Notice that this figure takes up the full page width.

5.1 Synchronous

5.2 Asynchronous

5.3 DC

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum

gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

5.4 Tables

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

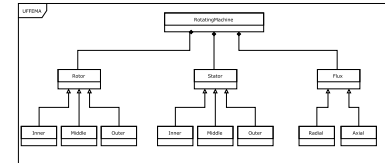


Figure 5.2: This is a margin figure. The helix is defined by $x = \cos(2\pi z)$, $y = \sin(2\pi z)$, and $z = [0, 2.7]$. The figure was drawn using Asymptote (<http://asymptote.sf.net/>).

This is a random margin note. Notice that there isn't a number preceding the note, and there is no number in the main text where this note was written. Use sidenote to use a number.

Strain	Growth Media				
	1	2	3	4	5
GDS1002	0.962	0.821	0.356	0.682	0.801
NWN652	0.981	0.891	0.527	0.574	0.984
PPD234	0.915	0.936	0.491	0.276	0.965
JSB126	0.828	0.827	0.528	0.518	0.926
JSB724	0.916	0.933	0.482	0.644	0.937
Average Rate	0.920	0.882	0.477	0.539	0.923

Table 5.1: Table caption text

5.5 Section 2

5.5.1 Subsection 1

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

5.5.2 Subsection 2

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam

euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consecetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

6

Material

Bibliography

K.T. Chau. *Electric Vehicle Machines and Drives: Design, Analysis and Application*. Wiley, first edition, 2015. ISBN 978-1-118-75252-4.

A.J. Pina Ortega. *EMANFES: Electric Machine Analysis with Finite Element Solvers, Reference Manual*. 2018a.

A.J. Pina Ortega. *EMANPY: Electric Machine Analysis with Python, Reference Manual*. 2018b.

Index

citation, [11](#)

license, [2](#)