UFFEMA DEVELOPERS

UFFEMA:
Unified Framework for Electric Machine Analysis

REFERENCE MANUAL VERSION 0.1

A. J. Piña Ortega

November 23, 2018

POWERED BY UFFEMA DEVELOPERS

https://github.com/ajpina/uffema

First printing, November 2018

Contents

List of Figures

List of Tables

Introduction

The **U**nified **F**ramework **F**or **E**lectric **M**achine **A**nalysis (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.

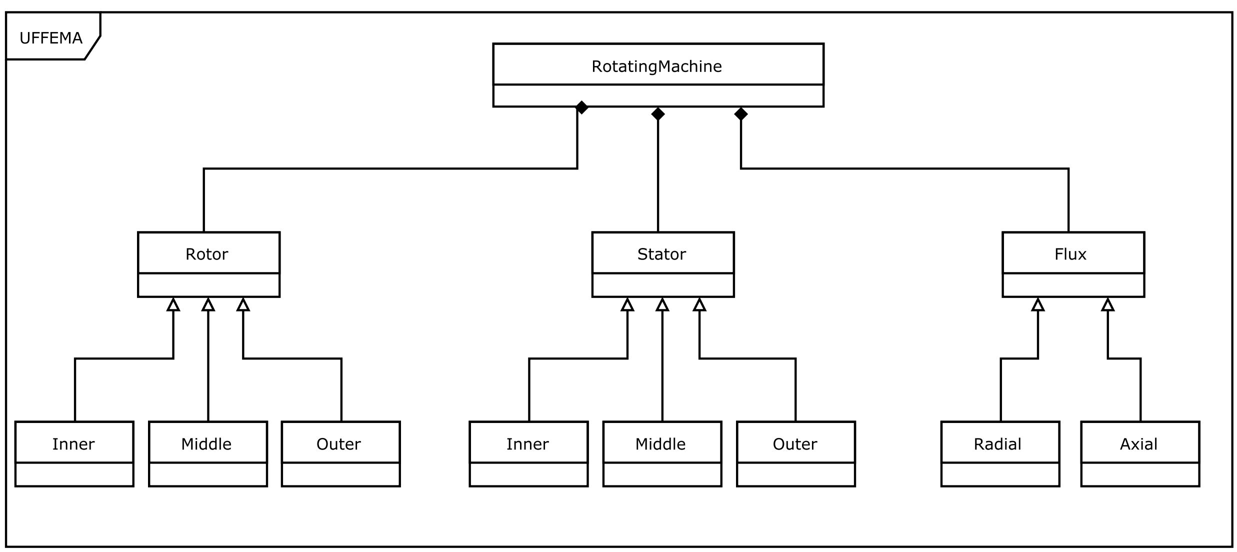

Figure 1: Composition of rotating electric machines

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.

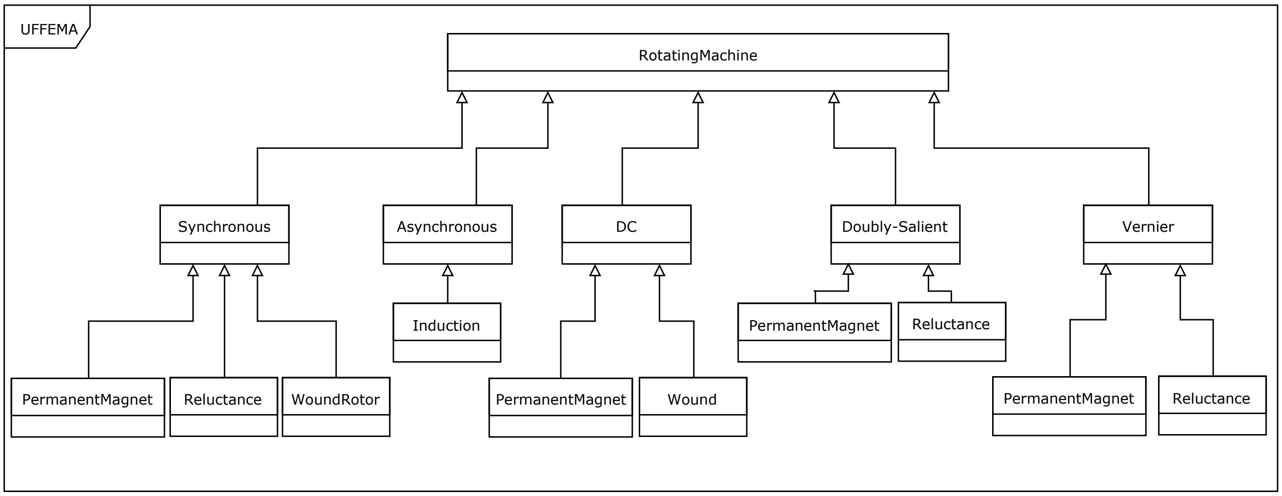

Figure 2: Classification of rotating electric machines

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 EMANPY[1] and EMANFES[2].

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 2, 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.

```
$ export PYTHONPATH=$PYTHONPATH:<directory>/uffema
$ cd uffema/tests
$ python 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.

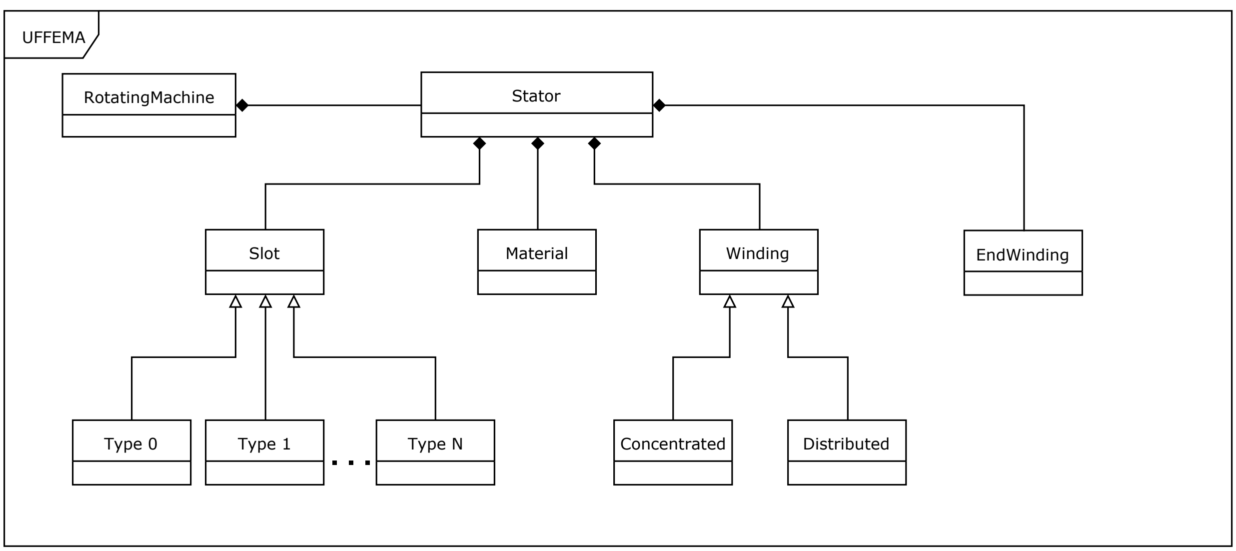

Figure 2.1: Stator architecture.

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`: *`String`* — Type of stator, for instance, *standardouter*, *standardinner*, etc.

`oSr`: *`Float`* — Stator outer radius.

`iSr`: *`Float`* — Stator inner radius.

`Ns`: *`Integer`* — Number of slots.

`Sl`: *`Float`* — Length of stator stack.

`slots`: *`Object`* — Settings for stator slots according to object detailed in chapter 3.

`material`: *`Object`* — Settings for stator material according to object detailed in chapter 6.

`winding`: *`Object`* — Settings for stator windings according to object detailed in chapter 4.

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

```
{
     "machine" : {
        "type" : String,
        "stator" : {
            "type" :  String,
            "oSr" : Float,
            "iSr" : Float,
            "Ns" : Integer,
            "Sl" : Float,
            "slots" : {...},
            "material" : {...},
            "winding" : {...}
        },
}
```

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, **`SOpos`** 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, *type*0, *type*1, etc.

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

```
"slots" : {
    "type" :  String,
    "dimension": [
        {
            "SOpos" : Float,
            "Spos" : Float,
            ...
        },
        {
            "SOpos" : Float,
            "Spos" : Float,
            ...
        },
    ]
}
```

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.

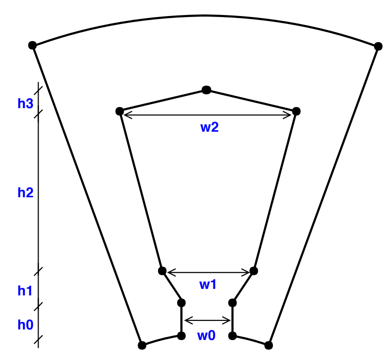

Figure 3.1: Parameters for slot type 0.

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 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`:	*`Float`*	Slot opening width.
`w1`:	*`Float`*	Wedge region width.
`w2`:	*`Float`*	Coil region width.
`h0`:	*`Float`*	Slot opening height.
`h1`:	*`Float`*	Wedge region height.
`h2`:	*`Float`*	Coil region height.
`h3`:	*`Float`*	Slot bottom height.

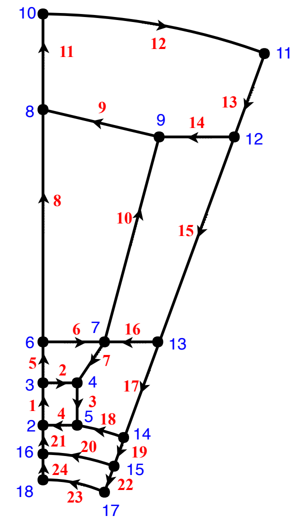

Figure 3.2: Type 0. Points, lines and loops.

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`:	*`Float`*	Slot opening width.
`w1`:	*`Float`*	Wedge region width.
`w2`:	*`Float`*	Coil region width.
`h0`:	*`Float`*	Slot opening height.
`h2`:	*`Float`*	Coil region height.

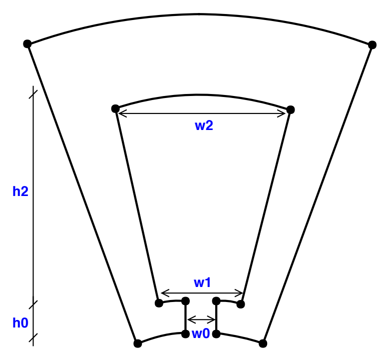

Figure 3.3: Parameters for slot type 1.

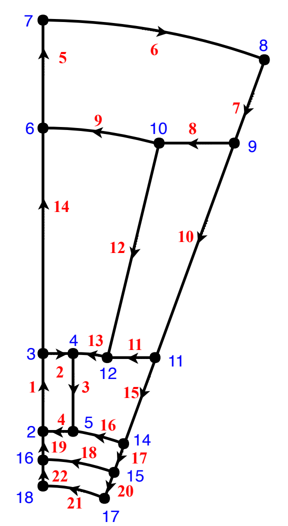

Figure 3.4: Type 1. Points, lines and loops.

3.3 *Type* 2

TO-DO: Implemented and documentation will be added shortly.

3.4 *Type* 3

TO-DO: Implemented and documentation will be added shortly.

3.5 *Type* 4

TO-DO: Implemented and documentation will be added shortly.

4 Windings

Windings consist of conductors that are conventionally inserted in the stator although there exist machines such as synchronous generator and induction motors, to name a few, where field windings and rotor bars can also be deemed as windings. Conductors are inserted into slots and overall coils can be arranged in multiple ways. In the current version of UFFEMA, windings can be distributed or concentrated and embedded in a layered manner. The following fields can be used to define the windings.

material: *Object* Settings for conductor material according to object detailed in chapter 6.

Layers: *Integer* Number of layers that will be arranged in the slots according with its **LayersType**.

LayersType: *String* Type of arrangement in the slots, i.e., *sidebyside* or *topbottom*, as long as layer number is even.

type: *String* Winding coils arrangement, i.e., *concentrated* or *distributed*.

Conn: *String* Type of winding connection, i.e., *wye* or *delta*.

NoPhases: *Integer* Number of phases.

Cseries: *Integer* Number of coils in series.

Cparallel: *Integer* Number of coils in parallel.

Cturns: *Integer* Number of turns per coil.

wih: *Integer* Number of wires in hand or stranded conductors.

condDiam: *Float* Conductor diameter.

Cpitch: *Integer* Coil pitch.

CM: *Array* Connection matrix, it is a list of parameters that define what conductors are inserted into the slots.

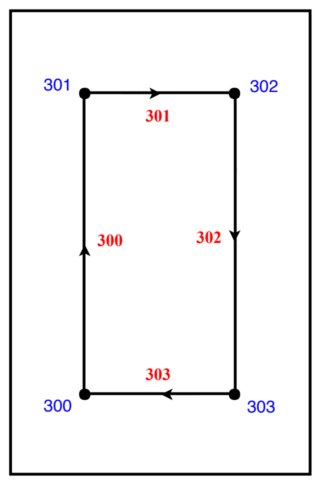

Figure 4.1: Two layers, side by side coils.

The connection matrix generally has dimension **Ns** by **NoPhases** for each coil side. There can be from zero to several coil sides within a slot. For instance, Figures 4.2-4.4 show one coil side which refers to split a slot into two parts side by side. Therefore, the conductor area in the figures are symmetrical with respect to the leftmost vertical line.

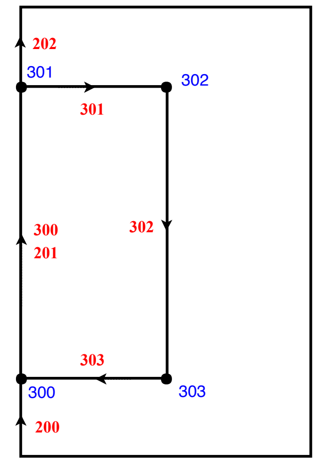

Figure 4.2: One layer coils.

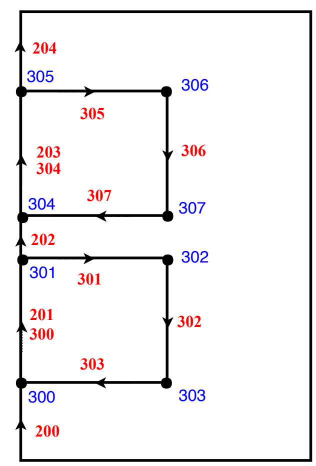

Figure 4.3: Two layer, top and bottom coils.

```
"winding" : {
    "material" :  {...},
    "Layers" : Integer,
    "LayersType" : String,
    "type" : String,
    "Conn" : String,
    "NoPhases" : Integer,
    "Cseries" : Integer,
    "Cparallel" : Integer,
    "Cturns" : Integer,
    "wih" : Integer,
    "condDiam" : Float,
    "Cpitch" : Integer,
    "CM" : {
        "CM1" : [
            {"Phase" : Value, ... },
            ...
          ],
          "CM2" : [
            {"Phase" : Value, ... },
            ...
          ]
     }
}
```

For the connection matrix to be defined, coil sides **CM#** need to be listed and each one consists of conductors belonging to the machine phases. Phases are **A** to **F** to denote one-phase to six-phases machine, respectively. Additionally, a value of 1 means current flowing positively perpendicular to the machine plane, namely +Z if it's assumed that geometry sits on XY-plane. On the contrary, -1 will indicate negative current going into -Z direction.

4.1 *One Layer*

Figure 4.2 shows points, lines and loops in order to build surfaces if the geometry is to be meshed. The conductor area results of merging the inner rectangle of both slot sides, whilst the outer one defines slot liner or insulation area. As a result, connection matrix requires only one coil side to be set.

4.2 *Two Layers - Side by Side*

Figure 4.1 shows points, lines and loops in order to build surfaces if the geometry is to be meshed. The conductor area will not be merged to its mirror, therefore the connection matrix needs the two coil sides since each conductor area might belong to different phase. The outer rectangle defines slot liner or insulation area.

4.3 *Two Layers - Top and Bottom*

Figure 4.3 shows points, lines and loops in order to build surfaces if the geometry is to be meshed. Analogous to previous configuration, with the difference that conductor areas are arranged vertically and will be merged to its mirror. Connection matrix and insulation area remain same as before.

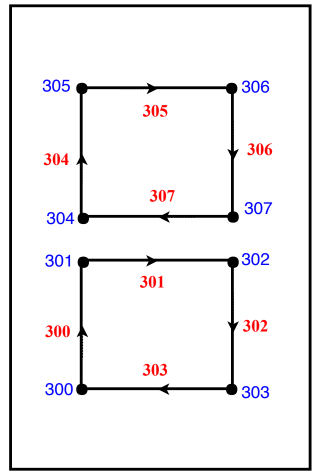

Figure 4.4: Four layer coils.

4.4 *Three Layers*

TO-DO: No implemented yet.

4.5 *Four Layers*

TO-DO: No implemented yet.

4.6 *Multiple Layers*

TO-DO: No implemented yet.

4.7 *Stranded Conductors*

TO-DO: No implemented yet.

5
Permanent Magnet Synchronous Machines

Rotor configuration mostly defines the type of electric machine, for instance, a permanent magnet synchronous machine consists of at least one rotor with magnets, pockets (none if SPM) and a material which rotor is made of. Figure 5.1 shows that UFFEMA defines a Permanent Magnet machines with a rotor that can be of type IPM or SPM. Several subcategories are implemented for each and set forth in the following sections.

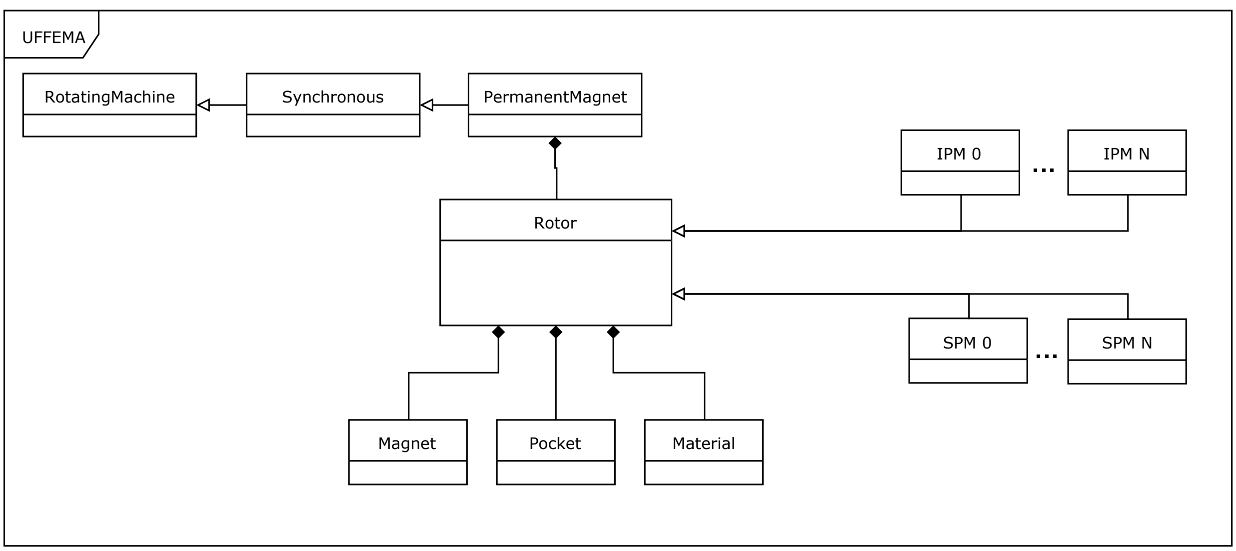

Figure 5.1: Configuration of permanent magnet synchronous machines.

5.1 Rotor

As can be seen in Figure 5.1, at least one rotor is found in a rotating machine and is generally characterized by having dimensions, pole pairs and material. The rest of parameters that describe magnet and magnet pockets features depends on each particular type of rotor configuration. The parameters recognized by UFFEMA are as follows,

type: *String* — Type of rotor, for instance, *spmo*, *ipmo*, etc.

oRr: *Float* — Rotor outer radius.

iRr: *Float* — Rotor inner radius.

pp: *Integer* — Number of pole pairs.

Rl: *Float* — Length of rotor stack.

magnets: *Object* — Settings for rotor magnets according to type of rotor.

material: *Object* — Settings for rotor material according to object detailed in chapter 6.

pockets: *Object* — Settings for magnet pockets according to type of rotor.

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

```
{
    "machine" : {
      "type" : String,
      "rotor" : {
          "type" :  String,
          "oRr" : Float,
          "iRr" : Float,
          "pp" : Integer,
          "Rl" : Float,
          "magnets" : {...},
          "pockets" : {...},
          "material" : {...}
      },
}
```

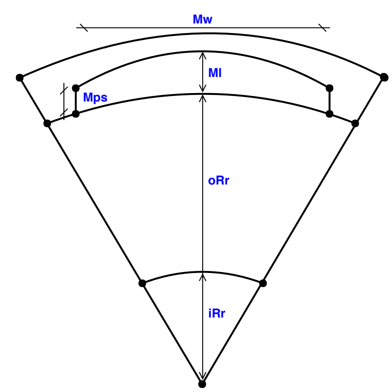

Figure 5.2: Parameters for rotor of surface permanent magnet motor type 0.

5.2 *Rotor Types*

Rotor type is tightly linked to magnet and pocket type. Hence, the next sections, that deal with type of permanent magnet rotors currently supported by UFFEMA, includes their own definition for its magnets and pockets.

5.2.1 *SPM Type* 0

The following parameters set forth magnet settings for surface permanent magnet machines with rotor type 0 (spm0), according to Figure 5.2 and 5.3. Notice that this geometry does not require to include any magnet pocket settings. The latter figure shows points, lines and loops in order to build surfaces if the geometry is to be meshed. Settings for rotor material and magnet material are according to object detailed in chapter 6. This type of rotor only supports *arc* magnets and these should be defined as follows:

type: *String* — Magnet type, i.e., *arc*.

magnetisation: *String* — Type of magnetisation, i.e., *parallel*, *radial*, *halbach*, etc.

material: *Object* — Settings according to object detailed in chapter 6.

dimension: *Array* — List of parameters that defines geometry for each magnet..

The length of the array **dimension** must be twice the value of **pp**, which allows to define an asymmetrical geometry being each magnet independently set. The parameters for magnets of type *arc* are:

Ml: *Float* — Magnet thickness, or magnet length measured radially.

Mw: *Float* — Magnet width.

Mps: *Float* — Pole shaping ratio, between 0 and 1. It scales accordingly the magnet length (**Ml**) at the sides.

iMr: *Float* — Radius of magnet inner arc, for *spm0*, this value should be the same as **oRr**.

delta: *Float* — Angle to tangentially shift magnets from their centre.

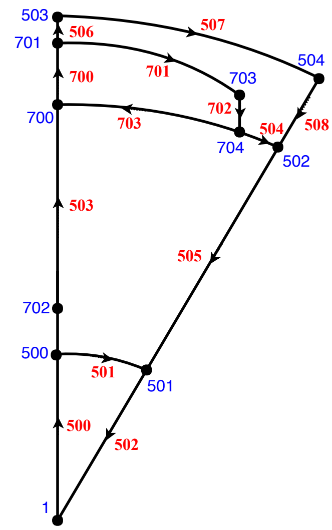

Figure 5.3: Rotor of surface permanent magnet motor type 0. Points, lines and loops.

5.3 *SPM Type* 1

TO-DO: Implemented and documentation will be added shortly.

5.3.1 *IPM Type* 0

TO-DO: Implemented and documentation will be added shortly.

5.3.2 *IPM Type* 1

TO-DO: Implemented and documentation will be added shortly.

6
Material

This object can be used to define either magnetic or non-magnetic materials. Most of its fields are explained below however it is noteworthy that not all of these are mandatory and should vary depending of the nature of material being defined.

`name`: *`String`* Material identification.

`BHcurve`: *`Array`* Values of B and H in order to define Induction curves.

`resistivity`: *`Float`* Material resistivity.

`Br`: *`Float`* Residual flux for hard magnetic materials..

`mur`: *`Float`* Relative permeability.

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

```
"material" : {
    "name" : String,
    "BHcurve" : [
            {"B" : Float, "H" : Float },
            ...
        ],
        "resistivity" : Float,
        "Br" : Float,
        "mur" : Float
}
```

7
Examples

Examples are available in the source code and located in the folder **uffema/tests**.

7.1 File: motor_0.msf

Surface permanent magnet motor.

This example shows a fractional slot concentrated winding machine (FSCW) with nine slots, six poles and surface permanent magnet rotor. It is a inner rotor configuration of type *spmo*, with Neodymium magnets, three-phase winding and steel M80065A.

```
{
  "machine" : {
    "type" : "spm",
    "stator" : {
      "type" : "standardouter",
      "oSr" : 42.5e-3,
      "iSr" : 18.98e-3,
      "Ns" : 9,
      "Sl" : 36.5e-3,
      "slots" : {
        "type": "type0",
        "dimension": [
          {
            "SOpos": 20.0,
            "Spos": 20.0,
            "h0": 0.89e-3,
            "h1": 1.5e-3,
            "h2": 18.5e-3,
            "h3": 1e-3,
            "w0": 2.55e-3,
            "w1": 6.4e-3,
            "w2": 16.4e-3
          },
          {
            "SOpos": 60.0,
            "Spos": 60.0,
            "h0": 0.89e-3,
            "h1": 1.5e-3,
            "h2": 18.5e-3,
            "h3": 1e-3,
            "w0": 2.55e-3,
            "w1": 6.4e-3,
            "w2": 16.4e-3
          },
          {
```

```
          "SOpos": 100.0,
          "Spos": 100.0,
          "h0": 0.89e-3,
          "h1": 1.5e-3,
          "h2": 18.5e-3,
          "h3": 1e-3,
          "w0": 2.55e-3,
          "w1": 6.4e-3,
          "w2": 16.4e-3
        },
        {
          "SOpos": 140.0,
          "Spos": 140.0,
          "h0": 0.89e-3,
          "h1": 1.5e-3,
          "h2": 18.5e-3,
          "h3": 1e-3,
          "w0": 2.55e-3,
          "w1": 6.4e-3,
          "w2": 16.4e-3
        },
        {
          "SOpos": 180.0,
          "Spos": 180.0,
          "h0": 0.89e-3,
          "h1": 1.5e-3,
          "h2": 18.5e-3,
          "h3": 1e-3,
          "w0": 2.55e-3,
          "w1": 6.4e-3,
          "w2": 16.4e-3
        },
        {
          "SOpos": 220.0,
          "Spos": 220.0,
          "h0": 0.89e-3,
          "h1": 1.5e-3,
          "h2": 18.5e-3,
          "h3": 1e-3,
          "w0": 2.55e-3,
          "w1": 6.4e-3,
          "w2": 16.4e-3
        },
        {
          "SOpos": 260.0,
          "Spos": 260.0,
          "h0": 0.89e-3,
          "h1": 1.5e-3,
          "h2": 18.5e-3,
          "h3": 1e-3,
          "w0": 2.55e-3,
          "w1": 6.4e-3,
          "w2": 16.4e-3
        },
        {
          "SOpos": 300.0,
          "Spos": 300.0,
          "h0": 0.89e-3,
          "h1": 1.5e-3,
          "h2": 18.5e-3,
          "h3": 1e-3,
```

```
            "w0": 2.55e-3,
            "w1": 6.4e-3,
            "w2": 16.4e-3
          },
          {
            "SOpos": 340.0,
            "Spos": 340.0,
            "h0": 0.89e-3,
            "h1": 1.5e-3,
            "h2": 18.5e-3,
            "h3": 1e-3,
            "w0": 2.55e-3,
            "w1": 6.4e-3,
            "w2": 16.4e-3
          }
        ]
      },
      "material" : {
        "name" : "stator_steel_M800_65A",
        "BHcurve" : [
          {"B" : 0.0, "H" : 0.0},
          {"B" : 0.1, "H" : 74.7},
          {"B" : 0.2, "H" : 97.5},
          {"B" : 0.3, "H" : 110},
          {"B" : 0.4, "H" : 120},
          {"B" : 0.5, "H" : 130},
          {"B" : 0.6, "H" : 140},
          {"B" : 0.7, "H" : 150},
          {"B" : 0.8, "H" : 162},
          {"B" : 0.9, "H" : 175},
          {"B" : 1.0, "H" : 190},
          {"B" : 1.1, "H" : 208},
          {"B" : 1.2, "H" : 227},
          {"B" : 1.3, "H" : 265},
          {"B" : 1.4, "H" : 366},
          {"B" : 1.5, "H" : 633},
          {"B" : 1.6, "H" : 1490},
          {"B" : 1.7, "H" : 3670},
          {"B" : 1.8, "H" : 7420},
          {"B" : 1.9, "H" : 13000},
          {"B" : 2.0, "H" : 21000},
          {"B" : 2.1, "H" : 34000},
          {"B" : 2.2, "H" : 55000},
          {"B" : 2.3, "H" : 88000},
          {"B" : 2.4, "H" : 140000},
          {"B" : 2.5, "H" : 220000},
          {"B" : 2.6, "H" : 349000},
          {"B" : 2.7, "H" : 550000},
          {"B" : 2.8, "H" : 860000},
          {"B" : 2.9, "H" : 1350000},
          {"B" : 3.0, "H" : 2200000},
          {"B" : 4.0, "H" : 60000000}
        ],
        "resistivity" : 1e-4
      },
      "winding" : {
        "material" : {
          "name" : "copper",
          "BHcurve" : null,
          "resistivity" : 1.724e-8
        },
```

```
        "Layers" : 2,
        "LayersType" : "sidebyside",
        "type" : "concentrated",
        "Conn" : "wye",
        "NoPhases" : 3,
        "Cseries" : 1,
        "Cparallel" : 3,
        "Cturns" : 22,
        "wih" : 1,
        "condDiam" : 1.52e-3,
        "Cpitch" : 1,
        "CM" : {
          "CM1" : [
            {"A" : 0, "B" : 1, "C" : 0, "D" : null, "E" : null, "F" : null },
            {"A" : 0, "B" : 0, "C" : 1, "D" : null, "E" : null, "F" : null },
            {"A" : 1, "B" : 0, "C" : 0, "D" : null, "E" : null, "F" : null },
            {"A" : 0, "B" : 1, "C" : 0, "D" : null, "E" : null, "F" : null },
            {"A" : 0, "B" : 0, "C" : 1, "D" : null, "E" : null, "F" : null },
            {"A" : 1, "B" : 0, "C" : 0, "D" : null, "E" : null, "F" : null },
            {"A" : 0, "B" : 1, "C" : 0, "D" : null, "E" : null, "F" : null },
            {"A" : 0, "B" : 0, "C" : 1, "D" : null, "E" : null, "F" : null },
            {"A" : 1, "B" : 0, "C" : 0, "D" : null, "E" : null, "F" : null }
          ],
          "CM2" : [
            {"A" : -1, "B" : 0, "C" : 0, "D" : null, "E" : null, "F" : null },
            {"A" : 0, "B" : -1, "C" : 0, "D" : null, "E" : null, "F" : null },
            {"A" : 0, "B" : 0, "C" : -1, "D" : null, "E" : null, "F" : null },
            {"A" : -1, "B" : 0, "C" : 0, "D" : null, "E" : null, "F" : null },
            {"A" : 0, "B" : -1, "C" : 0, "D" : null, "E" : null, "F" : null },
            {"A" : 0, "B" : 0, "C" : -1, "D" : null, "E" : null, "F" : null },
            {"A" : -1, "B" : 0, "C" : 0, "D" : null, "E" : null, "F" : null },
            {"A" : 0, "B" : -1, "C" : 0, "D" : null, "E" : null, "F" : null },
            {"A" : 0, "B" : 0, "C" : -1, "D" : null, "E" : null, "F" : null }
          ]
        }
      }
    },
    "rotor" : {
      "type" : "spm0",
      "oRr" : 15.2e-3,
      "iRr" : 6e-3,
      "pp" : 3,
      "Rl" : 36.5e-3,
      "magnets" : {
```

```
        "type" : "arc",
        "magnetisation" : "parallel",
        "material" : {
          "name" : "NdFeB32",
          "BHcurve" : null,
          "resistivity" : 1e-6,
          "Br" : 1.21,
          "mur" : 1.071
        },
        "dimension" : [
          {
            "Ml" : 2.98e-3,
            "Mw" : 13.6e-3,
            "Mps" : 1.0,
            "iMr" : 15.2e-3,
            "delta" : 0
          },
          {
            "Ml" : 2.98e-3,
            "Mw" : 13.6e-3,
            "Mps" : 1.0,
            "iMr" : 15.2e-3,
            "delta" : 0
          },
          {
            "Ml" : 2.98e-3,
            "Mw" : 13.6e-3,
            "Mps" : 1.0,
            "iMr" : 15.2e-3,
            "delta" : 0
          },
          {
            "Ml" : 2.98e-3,
            "Mw" : 13.6e-3,
            "Mps" : 1.0,
            "iMr" : 15.2e-3,
            "delta" : 0
          },
          {
            "Ml" : 2.98e-3,
            "Mw" : 13.6e-3,
            "Mps" : 1.0,
            "iMr" : 15.2e-3,
            "delta" : 0
          },
          {
            "Ml" : 2.98e-3,
            "Mw" : 13.6e-3,
            "Mps" : 1.0,
            "iMr" : 15.2e-3,
            "delta" : 0
          }
        ]
      },
      "material" : {
        "name" : "rotor_steel_M800_65A",
        "BHcurve" : [
          {"B" : 0.0, "H" : 0.0},
          {"B" : 0.1, "H" : 74.7},
          {"B" : 0.2, "H" : 97.5},
          {"B" : 0.3, "H" : 110},
```

```
          {"B" : 0.4, "H" : 120},
          {"B" : 0.5, "H" : 130},
          {"B" : 0.6, "H" : 140},
          {"B" : 0.7, "H" : 150},
          {"B" : 0.8, "H" : 162},
          {"B" : 0.9, "H" : 175},
          {"B" : 1.0, "H" : 190},
          {"B" : 1.1, "H" : 208},
          {"B" : 1.2, "H" : 227},
          {"B" : 1.3, "H" : 265},
          {"B" : 1.4, "H" : 366},
          {"B" : 1.5, "H" : 633},
          {"B" : 1.6, "H" : 1490},
          {"B" : 1.7, "H" : 3670},
          {"B" : 1.8, "H" : 7420},
          {"B" : 1.9, "H" : 13000},
          {"B" : 2.0, "H" : 21000},
          {"B" : 2.1, "H" : 34000},
          {"B" : 2.2, "H" : 55000},
          {"B" : 2.3, "H" : 88000},
          {"B" : 2.4, "H" : 140000},
          {"B" : 2.5, "H" : 220000},
          {"B" : 2.6, "H" : 349000},
          {"B" : 2.7, "H" : 550000},
          {"B" : 2.8, "H" : 860000},
          {"B" : 2.9, "H" : 1350000},
          {"B" : 3.0, "H" : 2200000},
          {"B" : 4.0, "H" : 60000000}
        ],
        "resistivity" : 1e-4
      },
      "init_pos" : -30.0
    }
  }
}
```

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