

Alex Aaltonen

**COUPLED FEM/SIMULINK MODELING OF
TRANSVERSAL FLUX SYNCHRONOUS
MACHINES WITH CORE LOSSES BY LOOKUP
TABLES**

Master's Thesis
Faculty of Information Technology and Communication Sciences
Examiners: Assoc. Prof. Paavo Rasilo
Joonas Vesa
June 2023

ABSTRACT

Alex Aaltonen: Coupled FEM/Simulink modeling of transversal flux synchronous machines with core losses by lookup tables

Master's Thesis

Tampere University

Master's Programme in Electrical Engineering

June 2023

In this thesis a novel approach for modelling multiphase transversal flux synchronous machines based on lookup tables is presented. The goal of the thesis is to develop a fast and reasonably accurate simulation model for the machine using MATLAB/Simulink. In addition, a decently fast way to approximate the iron core losses is included to the simulation model.

A simple 3D magnetostatic finite element simulation model of one phase of a transversal flux reluctance machine is made by using COMSOL Multiphysics software. The 3D finite element model is used to solve series of static simulations to produce the data for the lookup tables used in the Simulink model. In addition, another finite element model of a permanent magnet transversal flux synchronous machine developed by VTT is used to demonstrate the iron core loss calculations. The COMSOL model is also used to validate the Simulink model by simulating the model in time domain and comparing the results to results from the Simulink model.

Important data that needs to be obtained by solving the static finite element simulation model are the stator winding flux linkage, electromagnetic torque and magnetic flux density distribution as a function of current and rotor angle. The data is then stored to the lookup tables to be used in the Simulink model. From the lookup tables, it is then possible to get the values for the flux linkage and torque with different combinations of current and rotor angle.

To achieve efficient and fast simulations, the time domain evaluation of the machine is done by using MATLAB/Simulink with the help of utilizing the results of the static finite element simulations. The model is built in a way that it is possible to change the number of machine phases by only changing one parameter. Also, the winding connection is made possible to be switched between star and delta connections by changing only one parameter.

The iron core loss calculations utilizing the lookup tables are also included in the Simulink model. Loss calculations are implemented to the Simulink model by utilizing the magnetic flux density data produced by the static finite element simulations. To make the simulations faster with the iron core losses, proper orthogonal decomposition is used to decrease the amount of stored data. The model is also made faster by identifying homogeneous regions from the flux density distribution. It also helps to decrease the amount of data stored.

As a result, it is demonstrated that the simulations with the developed Simulink model are significantly faster than solving a time domain 3D finite element simulation. Including the iron core loss calculations to the model make the simulations slower, but by utilizing the data reduction methods, the simulations remain decently fast with a slight decrease in simulation accuracy.

Keywords: core loss, lookup tables, proper orthogonal decomposition, synchronous machines, transversal flux machines

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

TIIVISTELMÄ

Alex Aaltonen: Hakutaulukoihin perustuva rautasydänhäviöt sisältävä transversaalivuotahtikoneiden yhdistetty FEM/Simulink mallinnus

Diplomityö

Tampereen yliopisto

Sähkötekniikan DI-ohjelma

Kesäkuu 2023

Tässä diplomityössä toteutetaan uudenlainen hakutaulukoita hyödyntävä lähestymistapa monivaiheisen transversaalivuotahtikoneen mallintamiseen. Työn tavoitteena on kehittää nopea simulaatiomalli koneen toiminnan simuloimiseen. Simulaatiomalli toteutetaan MATLAB/Simulink-ohjelmistolla. Simulaatiomalliin on lisäksi sisällytetty nopea tapa arvioida koneen rautasydänhäviöitä.

Simulink-mallin toiminta perustuu hakutaulukoihin. Vaaditut tiedot hakutaulukoihin saadaan ratkaisemalla sarja staattisia 3D elementtimenetelmään perustuvia simulaatioita. Hakutaulukoihin vaadittavat tiedot ovat staattorin käämivuo, vääntömomentti ja magneettivuon tiheysjakauma virran ja roottorin kulman funktiona. Kahta elementtimenetelmään perustuvaa simulaatiomallia hyödynnetään hakutaulukoihin tarvittavien tulosten laskemiseen. Molemmat mallit on toteutettu COMSOL Multiphysics -simulointiohjelmistolla.

Yksi käytetyistä simulaatiomalleista on tätä työtä varten kehitetty malli yksinkertaisesta yksivaiheisesta transversaalivuoreluktanssikoneesta. Reluktanssikoneen mallia käytetään Simulink-mallin toiminnan todentamiseen ja sen ominaisuuksien esittelyyn. Toinen käytetyistä koneista on VTT:n kehittämä kestopagneettitransversaalivuotahtikone. VTT:n konetta käytetään rautasydänhäviöiden laskennan esittelyyn.

Simulinkin käyttö aikatazon simulointien toteutukseen mahdollistaa hyvin nopeat simulaatiot. Simulink-malli on toteutettu niin, että sen vaihemäärän muuttaminen onnistuu vain yhtä parametria muuttamalla. Myös staattorin käämityksen kytkennän muuttaminen tähti- ja kolmiokytkennän välillä onnistuu vain yhtä parametria muuttamalla.

Simulink-malliin on sisällytetty myös hakutaulukoita hyödyntävät rautasydänhäviölaskennat. Ne on toteutettu hyödyntämällä elementtimetelmäsimulaatioista saatavaa magneettivuon tiheysjakaumaa. Häviölaskennan sisällyttäminen Simulink-simulaatioihin hidastaa simulaatioita. Simulaatioita nopeutetaan hyödyntämällä redusointimenetelmää nimeltään "proper orthogonal decomposition". Sen avulla voidaan vähentää tallennettavan ja käsiteltävän tiedon määrää. Mallia nopeutetaan myös etsimällä yhtenäisiä alueita magneettivuon tiheysjakaumasta, mikä myös vähentää tallennettavan tiedon määrää.

Tuloksina esitellään, että simulaatiot kehitetyllä Simulink-mallilla ovat huomattavasti nopeampia, kuin aikatasossa ratkaistut 3D elementtimenetelmäsimulaatiot. Rautasydänhäviöiden sisällyttäminen malliin hidastaa simulaatioita, mutta tiedon redusointimenetelmien avulla simulaatiot pysyvät kohtuullisen nopeina, kärsien hieman tarkkuudessa.

Avainsanat: hakutaulukko, rautasydänhäviöt, tahtikone, transversaalivuokone

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

PREFACE

This thesis was written for the Electromechanics research group, which is part of the Faculty of Information Technology and Communication Sciences in Tampere University. This thesis was a part of joint research project with VTT. The Academy of Finland is acknowledged for financial support (grant no. 330062). The purpose of this thesis is to develop a fast simulation model of synchronous machines, which also includes iron core loss calculations.

I would like to thank the leader of the Electromechanics research group, associate professor Paavo Rasilo, for being an examiner for this thesis and for giving me an opportunity to write this thesis. Thanks to postdoctoral research fellow Joonas Vesa for being the second examiner for this thesis. They have both helped me a lot during the process of writing this thesis.

I would also like to thank Jenni Pippuri-Mäkeläinen and Firdausa Ahmed from VTT for being part of the project and providing the COMSOL model of the permanent magnet machine to test my model. Finally, thanks to everyone else in the Electromechanics group for being very supportive and helpful while I worked there.

In Tampere, 15th June 2023

Alex Aaltonen

CONTENTS

1.	Introduction	1
2.	Theory about electrical machines	2
2.1	Basic information about electrical machines	2
2.2	Types of electrical machines	3
2.2.1	Induction machines	3
2.2.2	Synchronous machines	4
2.3	Modelled machine	6
2.4	Machine model used to demonstrate the loss calculations.	8
2.5	Equation models	9
2.5.1	Voltage equations	9
2.5.2	Winding connection	11
3.	Simulation methods	14
3.1	COMSOL model	14
3.1.1	Geometry	15
3.1.2	Mesh	16
3.1.3	Physics interface	16
3.1.4	Solvers	23
3.1.5	Exported data	24
3.2	MATLAB model	25
3.2.1	MATLAB script	25
3.2.2	Relation of current frequency and rotor speed.	30
3.2.3	Simulink model	32
3.2.4	Iron core losses	35
3.2.5	Proper orthogonal decomposition	37
3.2.6	Homogeneous regions	38
4.	Results	40
4.1	Validating the Simulink model	40
4.2	Effects of phase number and connection type	42
4.3	Iron core losses	44
5.	Conclusion	50
	References	51
	Appendix A: Simulink model	54

LIST OF SYMBOLS AND ABBREVIATIONS

AC	alternating current
DC	direct current
FE	finite element
FEM	finite element method
IM	induction machine
LUT	lookup table
PMM	permanent magnet machine
PMSM	permanent magnet synchronous machine
POD	proper orthogonal decomposition
RM	reluctance machine
RMS	root mean square
SM	synchronous machine
SRM	synchronous reluctance machine
TFM	transversal flux machine
TFRM	transversal flux reluctance machine
TFSM	transversal flux synchronous machine

1. INTRODUCTION

Transversal flux synchronous machines (TFSM) are becoming popular in various applications. For example, in generation of electricity, a direct drive wind turbine [1] could be implemented by using a TFSM. Transportation is starting to rely more on electric machines due to the objective to reduce the need for fossil fuels. TFSMs in transportation applications could be used, for example, as motors for electric vehicles [2], [3] or in a smaller scale as motors for electric bikes [4]. With all the increased need for electric machines in the industry, efficient simulation models are needed to help in their development.

Models based on lookup tables (LUT) can be used for fast time domain simulations of synchronous machines [5]. However, loss calculation in time domain has not yet been properly assessed. In [6], eddy-current losses were included in LUT-based simulation using precalculated data from 2D finite element (FE) simulations. LUT-based models allow faster simulations than corresponding time domain FE models, but require some results to be precalculated with static FE simulations.

In this thesis, a new simulation model for multiphase transversal flux synchronous machines based on LUTs is proposed. The model allows the number of phases and stator winding connection to be varied by only changing simulation parameters, without the need of re-computing the LUTs. In addition, a computationally efficient approach for fast time domain evaluation of the core losses in LUT-based simulation of synchronous machines is proposed. The models are demonstrated for a transversal flux reluctance machine in COMSOL Multiphysics and MATLAB/Simulink.

2. THEORY ABOUT ELECTRICAL MACHINES

In this chapter, a short introduction to different electrical machine concepts is given. Also, the machine considered in this thesis is explained in more detail and the equation models of the machine are introduced.

2.1 Basic information about electrical machines

Machines that utilize electromagnetic forces can be called electrical machines. They are used in many different tasks like electric power generation, transportation, and industrial applications. They are an important part of the modern society.

Two important groups of electrical machines are electric motors and electric generators. Electrical energy is converted to mechanical energy with electric motors and the other way around with generators. Also, many other devices like transformers, which are utilized to control the voltage levels in alternating current systems, are electrical machines. The type of electrical machines considered important for this thesis are electric motors.

The structure of electrical motors and generators can be very similar. Many types of machines can be used as both motors and generators. However, the structure of different types of electrical motors and generators can be quite different depending on their operation principle.

Electrical motors and generators can either be rotational or linear. Rotational motors are used to convert electrical energy into rotational motion and the other way around in generator use. Linear motors are used to convert electrical energy into linear motion mechanical energy [7] and the other way around with linear generators.

In this thesis, only rotational motors are considered. Therefore, the machine parts and types are described from the point of view of rotational machines. In rotating machines, there are common parts that can be found in most types of rotating machines. The parts are the rotor, the stator, and the windings.

The rotor is the moving part of the electrical machine and in rotating machines it is rotating. The stator is the stationary part of the electrical machine. The gap between the stator and the rotor is called an air gap. The air gap length can have big effect on the operation of the machine depending on the machine type.

Windings or coils are sets of conductors that are used to conduct the current and they are located in the stator. The stator and the rotor cores are often made of a ferromagnetic material and the winding is constructed of wires that are good electrical conductors, like copper wires.

The mentioned parts are usually found in electrical motors and generators, especially in rotating machines. However, the placement and the shape of these parts can be very different in different types of machines and even in same type, but differently structured machines.

2.2 Types of electrical machines

To create understanding about electrical machines, some of the most common electrical motor types are shortly introduced. Because of the subject of the thesis is about electric motors, mostly motors are considered as electrical machines in this thesis. However, many electric motors mentioned here can also operate as generators and some examples of generator use are also given.

One important way to classify electrical motors is by the type of current they are driven by. This way two categories of electrical motors can be identified: alternating current (AC) motors and direct current (DC) motors. In this thesis, only AC motors are considered and DC motors are not discussed.

AC motors are supplied with AC current. The current is supplied to the windings on the stator, which generates a rotating magnetic field. The rotation of the magnetic field is then utilized to force the rotor into rotating motion. AC motors can be further split into two main categories, which are synchronous and induction motors.

2.2.1 Induction machines

Induction machines (IM) are applicable as both motors and generators. Induction motors are also called asynchronous motors [8, p. 172]. That is because the the rotor is rotating at different speed than the stator field. In motor operation, the rotor is rotating a little slower than the stator field.

The different rotating speeds of the rotor and the stator field is needed for the IM to produce any torque. The operation of the IM is based on the electromagnetic induction. The difference in the rotating speeds causes the stator field to be changing relative to the rotor, which then causes current to be induced to the rotor windings. A magnetic field is then created by the induced current in the direction to oppose the changing current in the windings of the rotor. The interaction between the stator field and the rotor magnetic field then causes the rotor to experience torque to the direction of the stator field rotation.

It is possible to implement the rotor of the IM in different ways. The main ways are squirrel cage rotor and wound rotor [8, p. 168]. The wound rotor has windings, similar to the stator, often built from copper wires. The squirrel cage rotor has longitudinal conductive bars, that are connected to a ring on each end creating a cage like structure.

The wound rotor offers a possibility to control the resistance of the rotor circuit after the rotor is constructed, which is not possible with the squirrel cage rotor [9, p. 155]. However, squirrel cage rotors are simpler in their construction. This usually makes them cheaper and more reliable option than the wound rotor [9, p. 156].

2.2.2 Synchronous machines

Synchronous machines (SM) are commonly used in both generator and motor uses [8, p. 81]. They get their name from the fact that the rotational speed of the rotor is synchronized with the supply current. This differs from the asynchronous machines where the rotor speed and the rotation speed of the stator field, which is determined by the supply current frequency, need to be different.

There are different synchronous machine types, which can be used in many different applications. For example, synchronous motors can be used in electrical vehicles as in [10], [11] and [12]. Synchronous generators can be used, for example, in wind turbines as in [13] and [14].

Advantages of synchronous motors are constant speed operation and high operating efficiency [15, p. 554-555]. Constant speed operation means that the speed of the rotor depends only on the frequency of the supply current, as long as the load is not too big to cause the motor to fall out of synchronization. The high operating efficiency can be achieved especially with machines like permanent magnet synchronous machines [16, p. 223].

The types of SMs introduced here in more detail are permanent magnet machines (PMM) and reluctance machines (RM). PMMs are introduced, because they are an important concept when discussing about transversal flux machines and the machine designed by VTT, used to demonstrate the iron core losses, is a PMM. RMs are introduced, because the example machine modelled in this thesis, and used to validate the Simulink model, works with the reluctance machine principle.

2.2.2.1 Permanent magnet machines

In permanent magnet synchronous machines (PMSM), permanent magnets are used for the magnetization of the rotor. The permanent magnets create a constant magnetic field to the rotor and the AC current in the stator windings creates a rotating magnetic field to

the stator. The rotation of the rotor happens, because the rotor poles lock to the rotating stator magnetic field and the speed of the rotation is synchronized with the supply current frequency.

Using permanent magnets in the rotor of electrical machines can increase the efficiency and power density of the machine [17]. The high efficiency and high torque density of the PMSM makes it a good machine type for electrical vehicle applications as in [10], because of the varying torque need and the limitation of available energy.

Potential disadvantages of PMSMs are the cost and availability of the permanent magnets. They are problems especially when considering high performance machines where rare-earth permanent magnets are used [17, p. 54]. Cheaper and more easily available magnets, like ferrite magnets, do exist, but they do not offer as good performance as the rare-earth magnets [18].

2.2.2.2 Reluctance machines

The operation of reluctance machines is based on the reluctance concept [15, p. 185]. The magnetic flux produced in the stator tries to go into the rotor on a path with minimal reluctance. The reluctance is minimum at the point where the stator and rotor poles are aligned. When the stator and rotor poles are not aligned, the rotor will experience a torque that tries to rotate it to create a minimal reluctance path for the magnetic flux. The torque is zero when the poles are aligned.

Because the torque production of the RMs is based on the reluctance concept, the rotor of the RM can be constructed simply from a ferromagnetic material. There is no need for windings or magnets on the rotor of a machine that is based solely on the reluctance concept. In some cases, there can be, for example, magnets added to the rotor for assistance as is done in [19], but in principle no magnets are needed for the RM to produce torque. However, adding magnets to the rotor for assistance does increase the efficiency and the torque of the RM [20, p. 38].

The lack of magnets and windings simplifies the construction of rotor, because in the simplest case it can be constructed from a single material. However, to create a highly efficient RM, the rotor geometry needs to be carefully designed as is shown in [21]. The lack of magnets in the rotor also lowers the cost of the machine when compared to machines that utilize magnets on the rotor.

The main advantages of the RMs are robustness and low cost [20, p. 27]. There are also some quite limiting disadvantages with the RMs, that are low power factor [20, p. 33] and the inability for self-starting [22, ch. 37.3.6].

2.2.2.3 Transversal flux machines

Transversal flux machine (TFM) is one design to implement a synchronous machine. They are named by the direction of the magnetic flux. TFMs can be different types of synchronous machines like PMSMs or synchronous reluctance machines (SRM). In TFMs, the magnetic flux flows mostly to the direction that is transverse to the moving direction [23].

Other conventional and well known machine designs are radial flux machines [24] and axial flux machines [25]. In radial flux machines, the direction of the generated magnetic flux is perpendicular to the rotation axis. In axial flux machines, the direction of the generated magnetic flux is parallel to the rotation axis.

Rotational TFMs can be used in many different applications. In [2], a rotational TFM is proposed as an in-wheel drive for an electric vehicle. In [4], a rotational TFM is developed to be a motor for an electric bike. In [26], a TFM is designed for generator use in large scale direct-drive wind turbines.

Rotational TFMs can have big differences in their construction. The rotor can be inside or outside of the stator. In outer rotor construction, the rotor is outside of the stator. The TFM developed in [27] is an example of an outer rotor TFM. In inner rotor construction, the rotor is inside the stator. The example machine modelled for this thesis is an inner rotor TFM.

2.3 Modelled machine

The structure of the TFRM modelled for this thesis is very basic for simplicity. The simple structure makes it easier to develop the simulation models. For an actual electric machine, this design would most likely not be the most optimal, but for developing the simulation models for this thesis this design is good enough. The design is similar to the transversal flux machine shown in [28], but with less poles. One phase of the TFRM model is depicted in Figure 2.1.

Only the active parts of one machine phase are depicted in Figure 2.1. Active parts mean the parts that are a part of the magnetic circuit. Actual machine may also contain parts that are not considered to be a part of the magnetic circuit. Instead, they can be a part of its support structure.

The machine is an inner rotor motor, because the rotor is located on the inside and the stator is located on the outside. The stator winding is circular and depicted as the copper-colored ring in the stator of the example picture. The stator cores are the U-shaped iron blocks around the winding. The rotor cores are attached to the rotor. The stator and the rotor cores are presumed to be made of a ferromagnetic material, which is suitable

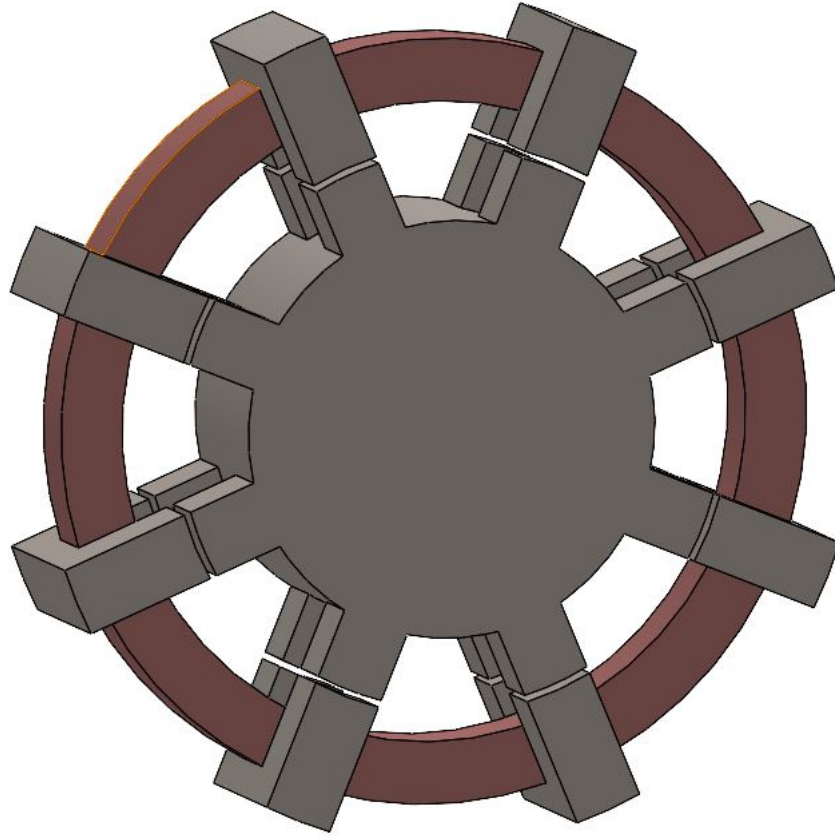


Figure 2.1. An example picture of one phase of the TFRM modelled for this thesis.

for reluctance machines. The stator winding is presumed to be constructed from copper wires.

The dimensions of the example model are not optimized in any way. In a realistic machine, the pole design would likely be much more complicated for more efficient torque production, instead of the very simple and basic design used in this example model.

To decrease the weight of the machine, the rotor would likely not be one big piece of solid iron, as in the example model. For example, the inner part of the rotor does not necessarily need to be part of the magnetic circuit, depending on the design of the machine. Then the inner part of the rotor could be made from other material than the parts that are part of the magnetic circuit. To reduce the amount of needed stator winding material, the rotor could also be located outside instead of inside.

Figure 2.1 depicts only one phase of the TFRM. For example, a three-phase machine would be constructed from three of the depicted phases and so on. One phase of the current is used for the excitation of one phase winding. Therefore, a machine with three phases needs a three-phase current to operate.

For the torque generation to be possible, rotor or stator of each phase needs to be phase shifted. In an actual machine, it would be beneficial to implement the phase shift to the stator instead of to the rotor for mechanical durability. However, in this model the phase

shift is implemented to the rotor for simplicity. The need for the phase shift is caused by the phase difference of the phase currents. The example seen in Figure 2.1 depicts a case where the rotor phase shift is 0° . An example of the case where the rotor phase shift is 22.5° can be seen in Figure 2.2.

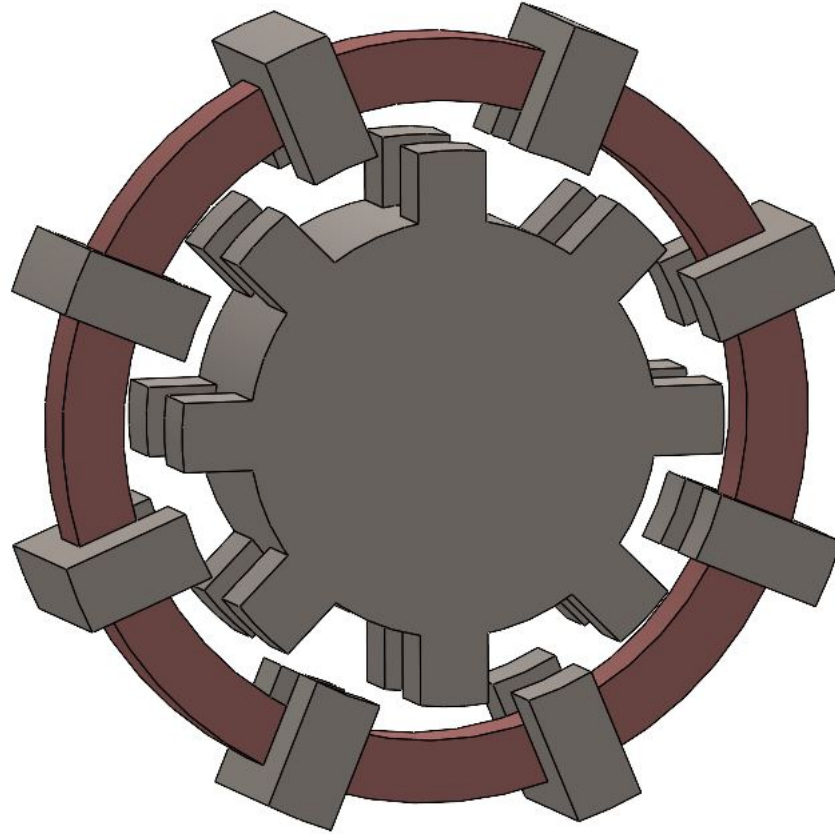


Figure 2.2. An example where the rotor shift is 22.5° .

In a three-phase case, the rotor phase shifts for the phases would be (0° , -15° and -30°). In a four-phase case, the phase shifts would be (0° , -11.25° , -22.5° and -33.75°). For this machine geometry, the phase shift is from the range of 0° to 45° . Because of symmetry, that range of angles contains all of the different positions of the rotor.

2.4 Machine model used to demonstrate the loss calculations

To demonstrate the computation of the iron core losses, a bit more complicated machine model developed by VTT was used. This was done to obtain more realistic results and, more importantly, to test the Simulink model with a different machine design. The results could have been demonstrated as well by using the reluctance machine model shown before, but this was a good opportunity to test the Simulink model with a different machine.

The more complicated machine is an outer rotor permanent magnet transversal flux synchronous machine. An example of the geometry of one full phase of the machine is demonstrated in Figure 2.3 and an example of one symmetry sector of the phase is

shown in Figure 2.4.

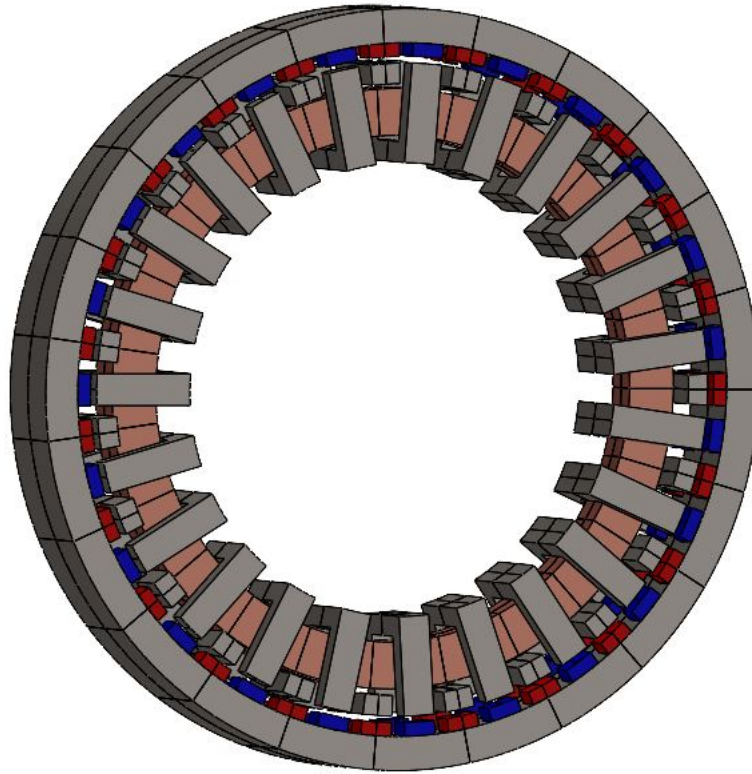


Figure 2.3. An example of one phase of the machine developed by VTT.

Only the active parts of the machine are shown in Figures 2.3 and 2.4. The stator consist of the coil and the stator iron cores and the rotor consist of the rotor iron core and the permanent magnets. The coil is displayed as copper colored, the iron cores are displayed as grey and the magnets are displayed as red and blue colors in the figures.

2.5 Equation models

To build the machine simulation model in Simulink, equations are needed to describe the operation of the machine. This section introduces the most important equations used to model the machine.

2.5.1 Voltage equations

To model the behavior of the machine in Simulink, a voltage equation is needed. The same voltage equation is applied as was used for the PMSM in [5]. The voltage equation for the machine is

$$\mathbf{u} = R_s \mathbf{i} + \frac{d\boldsymbol{\psi}}{dt}, \quad (2.1)$$

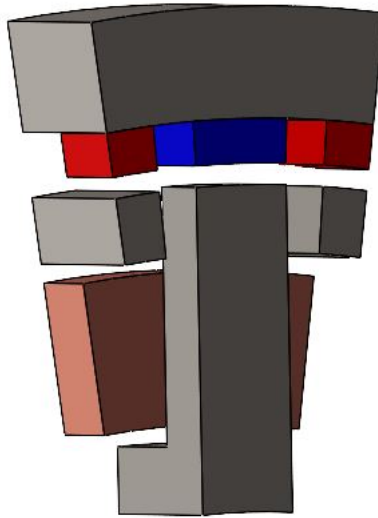


Figure 2.4. An example of symmetry sector of one phase.

where the vectors \mathbf{u} , \mathbf{i} and $\boldsymbol{\psi}$ include the voltages, currents and flux linkages of each phase. R_s is the resistance of the stator winding.

One possible way to handle the equation is to consider the flux linkage $\boldsymbol{\psi}$ and the rotor angle α_r as state variables. In this case, the voltage equation then becomes

$$\mathbf{u} = R_s \mathbf{i}(\boldsymbol{\psi}, \alpha_r) + \frac{d\boldsymbol{\psi}}{dt}. \quad (2.2)$$

When the voltage is supplied and the rotor angle is known, solving this form of voltage equation in Simulink requires a LUT containing the values of current as a function of flux linkage and rotor angle.

Another possibility is to consider the current and rotor angle as state variables. This way the voltage equation becomes a little more complicated and by the chain rule it can be written as

$$\mathbf{u} = R_s \mathbf{i} + \frac{\partial \boldsymbol{\psi}(\mathbf{i}, \alpha_r)}{\partial \mathbf{i}} \frac{d\mathbf{i}}{dt} + \omega_r \frac{\partial \boldsymbol{\psi}(\mathbf{i}, \alpha_r)}{\partial \alpha_r}, \quad (2.3)$$

where ω_r is the electrical rotational speed of the rotor.

Equation (2.3) can also be rearranged into a different form where the derivative of the current with respect to time is being solved. The equation can then be written as

$$\frac{d\mathbf{i}}{dt} = \left(\frac{\partial \psi(\mathbf{i}, \alpha_r)}{\partial \mathbf{i}} \right)^{-1} \left(\mathbf{u} - R_s \mathbf{i} - \omega_r \frac{\partial \psi(\mathbf{i}, \alpha_r)}{\partial \alpha_r} \right). \quad (2.4)$$

When using this equation in Simulink, the LUTs need to contain the partial derivatives of the flux linkage with respect to the rotor angle and the current. The LUTs are constructed from the data produced by static 3D FEM simulations and they are required for the Simulink model to work.

2.5.2 Winding connection

The stator windings of the electrical machine can be connected in so called delta or star connection. In the delta connection, the first end of one coil is connected to the second end of the other coil and so on. Line wires are connected to the connection points of the windings. With three phases, the delta connection can be depicted with the kind of triangular closed loop shown in the left side of Figure 2.5.

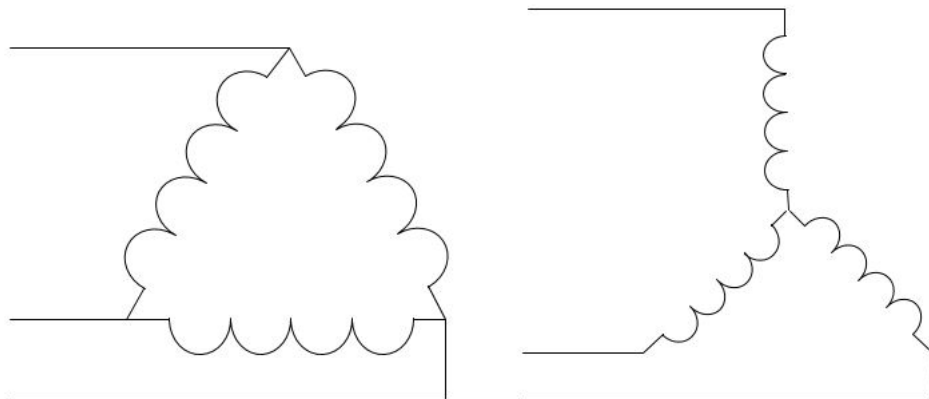


Figure 2.5. Delta and star connections with three phases.

In the star connection, the similar winding ends are connected to one point and the other ends are connected to the line wires creating a star type of pattern. An illustration of a three-phase star connection can be seen in the right side of Figure 2.5.

The middle point, where the windings are connected to each other in the star connection, can be called neutral point. Neutral wire can be connected to the neutral point, which would make a three-phase system to have four wires. With delta connection, it is not possible to use the neutral wire, because the neutral point does not exist.

The connection type has an effect on the relation between the line-to-line voltages and the phase voltages [15, p. 16]. The connection has an effect on the magnitude of the voltages and the phase angles of the voltages. The delta connection is the simpler case when the voltages are considered. The magnitudes of the line-to-line voltages and the

phase voltages can be considered equal as in

$$u_{LL} = u_{ph}, \quad (2.5)$$

where u_{LL} is the magnitude of the line-to-line voltage and u_{ph} is the magnitude of the phase voltage. Also, the phase angles of the line-to-line voltages and the phase voltages are equal with the delta connection.

The star connection is a more complicated case when the voltages are considered. In three-phase star connection, the magnitudes of the fundamental voltages can be expressed as

$$u_{LL} = \sqrt{3}u_{ph}, \quad (2.6)$$

which means that the magnitude of the phase voltage will be smaller than the magnitude of the line-to-line voltage. For this reason, the line-to-line voltage needs to be increased with the star connection, if the phase voltage is kept the same with both connections. There will also be a difference in the phase angles of the fundamental voltages. In three-phase star connection, the line-to-line voltages lead the corresponding phase voltages by 30° .

With the delta connection, the voltage equations presented earlier would be applicable as they are, but when using the star connection the voltage equations will require some modification using methods introduced in [29]. When a three-phase machine is connected with the star connection, the voltage equation is

$$\frac{d}{dt} \begin{bmatrix} i_a \\ i_b \end{bmatrix} = \left(\mathbf{K} \frac{\partial \psi_{abc}(i_{abc}, \alpha_r)}{\partial i_{abc}} \mathbf{K}^T \right)^{-1} \left(\mathbf{Q}u - \mathbf{K}R_s\mathbf{K}^T \begin{bmatrix} i_a \\ i_b \end{bmatrix} - \omega_r \mathbf{K} \frac{\partial \psi_{abc}(i_{abc}, \alpha_r)}{\partial \alpha_r} \right), \quad (2.7)$$

where the matrices \mathbf{K} and \mathbf{Q} are

$$\mathbf{K} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}, \quad (2.8)$$

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & -1 \end{bmatrix}. \quad (2.9)$$

The matrices \mathbf{K} and \mathbf{Q} for any number of phases are also introduced in [29] and they are

$$\mathbf{K} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & -1 \\ 0 & 1 & 0 & \dots & 0 & -1 \\ 0 & 0 & 1 & \dots & 0 & -1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix}, \quad (2.10)$$

$$\mathbf{Q} = - \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & 1 & 0 & \dots & 0 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 0 & 1 \end{bmatrix}, \quad (2.11)$$

where the matrix dimensions are $m - 1 \times m$, where m is the number of machine phases. Matrices (2.10) and (2.11) are applicable to cases where there are three or more phases. These matrices are not applicable for systems with one or two phases.

Systems with only one or two phases cannot be connected to star or delta connection, at least by the general definition of the connections. Therefore, the two-phase case is not considered in the Simulink simulations. The one phase case is only used to verify that the Simulink model works correctly, because only one phase of the machine is modelled with COMSOL Multiphysics.

3. SIMULATION METHODS

In this chapter, the simulation models are presented. The COMSOL and the Simulink models developed for this thesis are introduced and some of the theory behind the simulation features is presented.

3.1 COMSOL model

COMSOL Multiphysics is a simulation software that allows simulating physics-based models using finite element method (FEM). FEM is a method to numerically solve partial differential equations that can arise from different models, for example, from an electric machine. When solving a problem with FEM, the system is divided into smaller parts. These so-called finite elements are generated by constructing a mesh of the object in question.

In this thesis, COMSOL is used to solve series of static simulations of transversal flux synchronous machines. Simulations are carried out with different values for current and rotor position angle. The relevant quantities solved from COMSOL simulations are the flux linkage and the electromagnetic torque. In addition, for the iron core loss calculations, the flux density distribution is required. The results from COMSOL simulations are then stored in LUTs, to be used when solving time dependent behaviour of the machine with Simulink.

Time dependent simulations could also be done with COMSOL, but they would take a lot of time. One time dependent simulation with COMSOL can take multiple hours, while with Simulink they could be done in few seconds. Therefore, using Simulink instead of COMSOL for time dependent simulations makes it much more convenient to investigate, for example, the effects of different voltage values to the machine operation. Time dependent simulations were also solved, for the example machine model used for this thesis, to compare the results to the Simulink solutions. The details about how the COMSOL features considered in this chapter work and how they could be used can be found in the COMSOL documentation [30].

3.1.1 Geometry

The geometry of one phase of the modelled machine can be seen in Figure 2.1 shown earlier. The one phase geometry model example can be divided into eighth identical sectors. One sector can then be solved and from the results the results for the full phase can then be deduced. In COMSOL this can be done easily by using the sector symmetry. The main advantages of using the sector symmetry are that it decreases the simulation time significantly and it reduces the amount of data produced by the simulation. The modelled sector of the machine is shown in the left side in Figure 3.1.

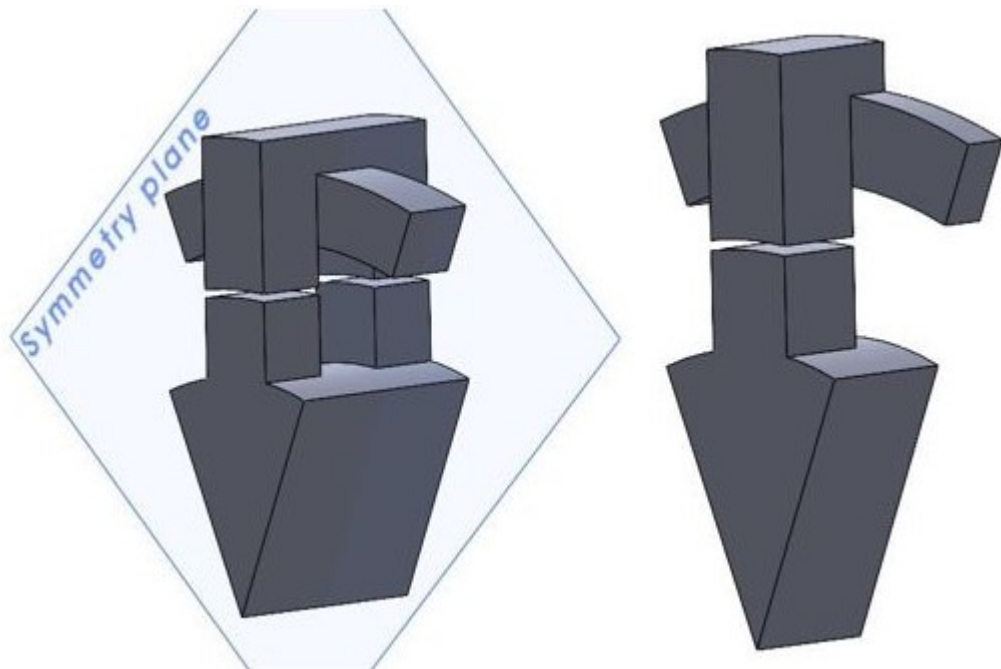


Figure 3.1. Symmetry sector and half of the symmetry sector of one machine phase.

As seen in the left side in Figure 3.1, the sector is symmetric with respect to the symmetry plane. For that reason, only half of the sector needs to be solved and then using these results the results of the full sector can be deduced. Illustration of the half of the symmetry sector can be seen in the right side in Figure 3.1.

Being able to solve the problem with only a half of the symmetry sector further reduces the computational burden and the amount of stored data. The half sector geometry depicted in the right side in Figure 3.1 is the geometry that is used for both stationary and time domain COMSOL simulations. The necessary simulation results are then scaled to represent the results of a full phase case. For example, the *Arkkiio Torque Calculation* feature takes the number of sectors into consideration when calculating the torque. Then the torque results just need to be multiplied by two, to get the torque of the full phase.

3.1.2 Mesh

The mesh was generated using the *Physics-controlled mesh* option in COMSOL. It lets the physics interface to determine the meshing sequence. Another option would be to create the mesh manually by using the *User-controlled mesh*, but for this case the *Physics-controlled mesh* seemed produce good results. A more complicated geometry might need the manually constructed mesh to work reliably. The mesh used for the model is depicted in Figure 3.2.

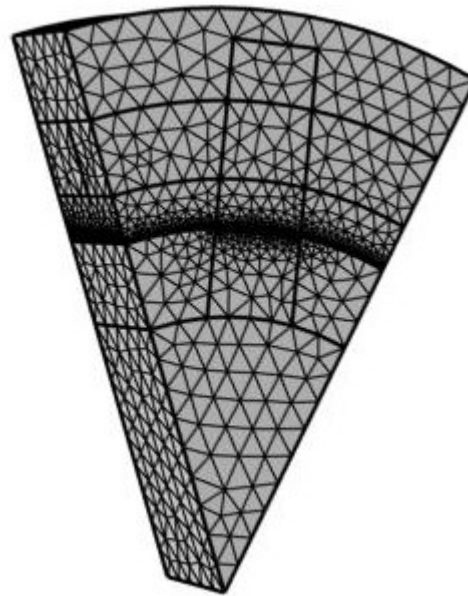


Figure 3.2. *The mesh used for the model.*

As seen in Figure 3.2, the mesh has different sized elements in different parts of the model. The less important parts of the model are constructed from much coarser mesh than the more important parts. The less important parts are, for example, the air part above the stator, and the inner part of the rotor. The most important parts, that require finer mesh, are in this case the air gap, and the tips of the poles.

The rotation of the rotor mesh is implemented by using the *Rotating Domain* feature under the *Moving Mesh*. This makes it possible to set the rotation speed or the rotation angle. For the stationary simulation, the *User Defined Rotation* was used to set the rotation angle for each static simulation using the *Parametric Sweep*.

3.1.3 Physics interface

The physics interface used for the COMSOL model is the *Rotating Machinery, Magnetic interface*. It can be used for both stationary and time domain analysis of rotating electric

motors and generators. In this thesis, 3D models are solved, but 2D models would also be possible to be solved with this physics interface. It utilizes the magnetic vector potential and the magnetic scalar potential as the dependent variables to formulate Maxwell's equations, which can then be solved to solve the system.

3.1.3.1 Mixed formulation

Two methods are used by the *Rotating Machinery, Magnetic interface* to solve Maxwell's equations. The methods are magnetic scalar potential and magnetic vector potential. The boundary between the two methods needs a *Mixed Formulation Boundary* feature between them. It is a default feature and added automatically to all required boundaries.

The magnetic vector potential can be applied by using the *Ampère's Law* feature. It allows to add Ampère's law to the magnetic field. Ampère's law describes mathematical connection between the electric current flowing in a conductor and the magnetic field induced around the conductor. It can be described as an equation in differential form as

$$\nabla \times \mathbf{H} = \mathbf{J}, \quad (3.1)$$

where $\nabla \times \mathbf{H}$ is the curl of the magnetic field strength and \mathbf{J} is the current density.

The *Ampère's Law* feature needs to be used in domains that are surrounding a domain with current density. Therefore, it needs to be used in the air domains in the stator and in the stator core domain that surround the stator winding domain. In addition, when *External Current Density* feature is used to generate the current density to the coil, the *Ampère's Law* feature is used in the coil domain.

The settings of the *Ampère's Law* feature differ slightly depending on the domain it is used in. For the air and coil domains, the relative permeability is set as the magnetization model. It sets the constitutive relation

$$\mathbf{B} = \mu_0 \mu_r \mathbf{H}, \quad (3.2)$$

where \mathbf{B} is the magnetic flux density, μ_0 is the permeability of the vacuum, μ_r is the relative permeability of the air which is set to 1 and \mathbf{H} is the magnetic field strength.

For the stator core domain, the B-H curve is set as the magnetization model. It sets the constitutive relation

$$\mathbf{B} = f(\|\mathbf{H}\|) \frac{\mathbf{H}}{\|\mathbf{H}\|}, \quad (3.3)$$

which means that a B-H curve, defined in the material properties, is used to get the relation between \mathbf{B} and \mathbf{H} . The rest of the settings were set to their default values and the rest of the properties were set to be taken from the default material settings in COMSOL.

The magnetic scalar potential can be applied by using the *Magnetic Flux Conservation* feature. It can be used to determine the magnetic field strength in a situation where there are no free currents effecting the domain. Because there is no free current, it can be presumed that

$$\nabla \times \mathbf{H} = 0. \quad (3.4)$$

Because of the requirement of no free currents, the *Magnetic Flux Conservation* feature can be used in the rotor air and iron core domains and in the air gap domain. Settings of the features are default, except for the magnetization model on the constitutive relation B-H. For air domains, the relative permeability shown in (3.2) is used. For iron core domains, the B-H curve shown in (3.3) is used.

3.1.3.2 Coil

The coil is implemented differently in the static simulation model and in the time domain simulation model. In the static simulation model, the coil excitation is implemented by using the *External Current Density* feature, while in the time domain model it is implemented by using the *Coil* feature. Different methods were used, because the *Coil* feature allows an easy way for voltage excitation for the time domain simulation.

An externally generated current density \mathbf{J}_e can be added by using the *External Current Density* feature. In the equation for the Ohm's law, the external current density is applied as

$$\mathbf{J} = \sigma \mathbf{E} + \mathbf{J}_e, \quad (3.5)$$

where \mathbf{J} is the overall current density, σ is the electrical conductivity and \mathbf{E} is the electric field.

In this case, the magnitude for the external current density is defined by

$$J_e = \frac{N i_1}{a_{\text{coil}}}, \quad (3.6)$$

where N is the number of stator winding turns, i_1 is the current and a_{coil} is the surface area of the coil. The current is defined as

$$i_1 = \sqrt{2}I_{\text{rms}}k_i, \quad (3.7)$$

where I_{rms} is the root mean square (RMS) value of the rated current and k_i is a parameter used for the parametric sweep. With the parameter k_i , the magnitude of the current can be controlled during the sweep.

The *Coil* feature was used to model the coil in the time dependent simulation model. It allows modelling of coils of different shapes and sizes. It is also possible to use different excitation methods with the *Coil* feature.

To get the *Coil* feature working correctly, many settings need to be adjusted. The conductor model in this case needs to be chosen as the *homogenized multiturn*. It models a case where there is a bundle of tiny electrically insulated wires wound together. The coil of the TFRM analyzed in this model is circular, as seen in Figure 2.1, so the coil type setting can be chosen to be circular.

Either current or voltage can be used for the coil excitation with the *Coil* feature. In the Simulink model, the voltage is used as controlled input variable. The time dependent COMSOL simulation is used to verify the Simulink simulation results. Therefore, voltage needs to be used for the coil excitation in the time dependent COMSOL simulation. The coil voltage is defined as

$$U_{\text{coil}} = \sqrt{2}U_{\text{rms}}\cos(\omega t), \quad (3.8)$$

where U_{rms} is the RMS value of the coil voltage and ω is the electric angular frequency.

Because the conductor model is the *homogenized multiturn* and coil shape is circular, the *Coil Geometry* feature is needed. It is used to choose the direction of the coil wires and to specify required geometry variables.

3.1.3.3 Boundary conditions

It is necessary to define boundary conditions to solve the partial differential equations generated by the COMSOL simulation. The fact that the geometry of the simulated model is reduced to the geometry depicted on the right side in Figure 3.1, means that the boundary conditions need to be carefully selected, even on the boundaries of the symmetry sector. If the boundary conditions are not correct, the simulation will not produce correct results.

The *Mixed Formulation Boundary* feature is needed on a boundary between magnetic scalar potential and magnetic vector potential to impose continuity between them. This boundary condition is added automatically by the COMSOL to the boundaries where it is

needed. In this model, the *Mixed Formulation Boundary* feature is used on the boundary on the stator side of the air gap.

The default boundary condition set for the external boundaries is the *Magnetic Insulation* feature. It is added automatically to all the external boundaries where it is not overridden by any other boundary condition. In this model, it is used on the external boundaries highlighted with blue in Figure 3.3.

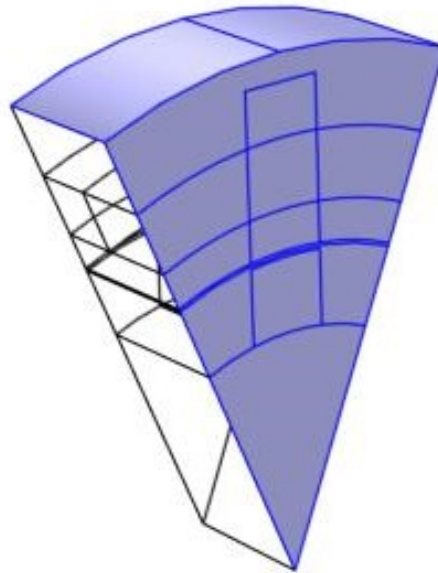


Figure 3.3. Boundaries where the *Magnetic Insulation* feature is applied.

As seen in Figure 3.3, the *Magnetic Insulation* feature is used on the external boundaries of the model. On boundaries with magnetic vector potential, it sets the tangential components of the magnetic vector potential to zero. On boundaries with magnetic scalar potential, it sets the normal component of the magnetic flux density to zero.

The *Periodic Condition* feature is used to choose between periodicity or antiperiodicity across boundaries. It is used on the boundaries that connect to the sectors defined by the sector symmetry. The boundaries where the feature is used are highlighted with blue in Figure 3.4.

The *Periodic Condition* sets how, for example, the vector potential is perceived through the boundary. If continuity is chosen, the potential will be equal on the source and destination sides. If antiperiodicity is chosen, a positive potential on the source side will be negative on the destination side. In this model, the continuity was used to ensure the continuity of the potential through the sectors.

The *Perfect Magnetic Conductor* feature sets the tangential component of the magnetic field to zero. It also sets the current density on the surface to zero. In this model, it is used on the interior boundaries of the domains where magnetic vector potential is used. The boundaries where it is used are highlighted with blue in Figure 3.5.

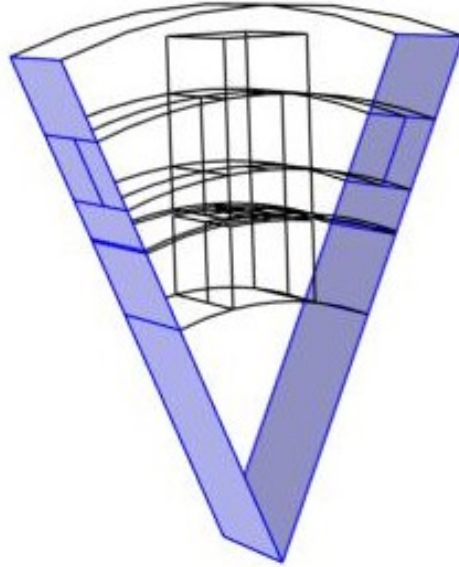


Figure 3.4. Boundaries where the Periodic Condition feature is applied.

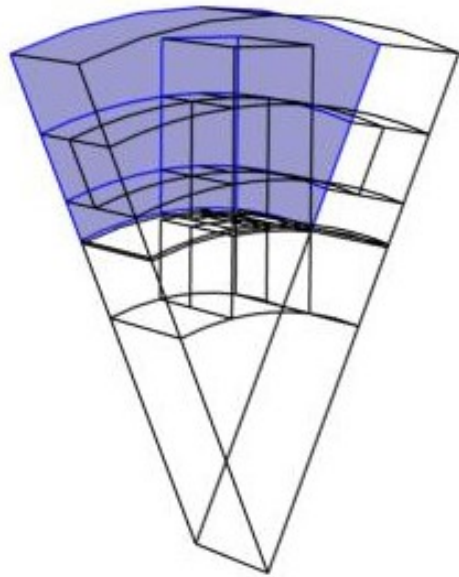


Figure 3.5. Boundaries where the Perfect Magnetic Conductor feature is applied.

By using the *Perfect Magnetic Conductor* feature, it can be ensured that the current flowing in the coil is mirrored to the other side of the border and that the current flow remains as tangential. It also ensures that the magnetic field is only allowed to point to the normal direction through the boundary.

The *Magnetic Scalar Potential* feature for the boundary condition can be used with domains where the *Magnetic Flux Conservation* feature is used. The boundaries where it is used are depicted in Figure 3.6.

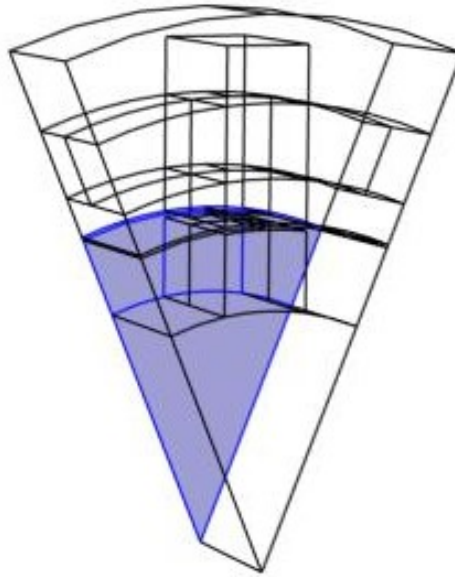


Figure 3.6. The boundaries where the Magnetic Scalar Potential feature is utilized.

The *Magnetic Scalar Potential* feature is utilized to set Dirichlet boundary condition that sets the magnetic scalar potential to a constant value on the boundary. In this case, the magnetic scalar potential was set to zero.

3.1.3.4 Torque calculation

The torque calculation is done using the *Arkio Torque Calculation* feature. Arkkio's method is based on the Maxwell stress tensor method and it is applicable for machines, which have an air gap with uniform thickness separating the rotating parts. According to the COMSOL documentation the torque is calculated by equation

$$T_{\text{ark}} = n_{\text{sect}} \int_{\Omega} \frac{F_{\phi} r}{r_{\text{out}} - r_{\text{in}}} dV, \quad (3.9)$$

where

$$F_{\phi} = \frac{B_r B_{\phi}}{\mu_0}, \quad (3.10)$$

and

$$\Omega : \{r_{\text{in}} < r < r_{\text{out}}\}, \quad (3.11)$$

where n_{sect} is the number of symmetry sectors, F_{ϕ} is the azimuthal force density, r is the

distance from the rotation center, r_{in} and r_{out} are the inner and outer radius of the air gap, B_r and B_ϕ are the radial and azimuthal components of the magnetic flux density and Ω is the integration domain. More detailed information about Arkkio torque calculation method can be found in [29].

3.1.4 Solvers

A *Stationary Solver* is used to get the data to develop the Simulink model and a *Time Dependent Solver* is used to verify the results of the Simulink model. Therefore, both solvers are discussed here.

3.1.4.1 Stationary model

When solving the static model, the *Parametric Sweep* is used to repeat the simulation multiple times with different values for the rotor angle and the current. Basically, it means that the static simulation is repeated for each combination of the rotor angle and the current.

The *Stationary Study Step* in COMSOL can be used when solving problems that contain no field variables changing over time. This works for the model in question because the idea is to solve the static state of the model with certain value for the current and the rotor angle.

The *Stationary Solver* and the added features are used mostly with their default settings. The features added to the *Stationary Solver* are *Direct*, *Advanced*, *Parametric*, and *Fully Coupled*. The *Parametric* feature is added together with the *Parametric Sweep Study* to handle the settings for the parameter stepping. The *Advanced* feature is added automatically when the *Stationary Study* is added and it is used to set advanced settings for the *Stationary Solver*.

The *Direct* feature is used to set the direct linear system solver settings. The *Iterative* feature, which handles iterative linear system solvers, would be a different option for the *Direct* feature. In this case, the *Direct* feature was chosen to be used.

The *Fully Coupled* feature is chosen as the solution method to use the fully coupled solution. Other option would be the *Segregated* feature, which controls the segregated solver. The difference between these two methods is that the fully coupled solution method solves the problem as one large system, while the segregated solution method divides the problem into smaller steps, which are then solved sequentially.

3.1.4.2 Time dependent model

The time domain solution of the COMSOL model is used to verify the results of the Simulink simulations. To solve problems where the field variables change over time, the *Time Dependent Study Step* needs to be used. It adds the *Time Dependent Solver* as a solver. There are certain default settings in the *Time Dependent Solver* that need to be changed, so that the simulation works reliably.

In the *Time Dependent Solver*, the setting controlling the solver steps was changed from free to manual. After that the time step size was set to a custom value, to make it easier to observe the behaviour of the solver, while debugging the time domain simulation.

The additional features used for the *Time Dependent Solver* are *Direct*, *Advanced* and *Fully Coupled*. They are also used for the *Stationary Solver*. The *Direct* and *Advanced* features are used with their default settings. The *Fully Coupled* feature needs a setting change for the simulation to run correctly. The setting *Nonlinear Method* is set to *Automatic (Newton)*.

3.1.5 Exported data

To utilize the data from the COMSOL simulations, it needs to be exported into a form that is suitable for MATLAB. The data from COMSOL can easily be exported into a text file, which can then be read with MATLAB. Large systems may provide large amount of data, which might cause issues with reading and saving the data on MATLAB. The data used in simulations in this thesis work did not cause any issues because of its size.

It is possible to export a lot of different variables from COMSOL. For the Simulink model introduced in this thesis, some of the variables are required and some of them are useful for presenting the results. The values of the rotor angle and the coefficient k_i of the current that are both used as parameters in the parametric sweep need to be exported to be able to create the LUTs with correct dimensions.

The values of current, flux linkage, and torque on every step of the parametric sweep also need to be exported from COMSOL. These are the data stored to the LUTs and because of that they are required for the operation of the Simulink model.

One important information to be included to the exported data, is the number of sectors that the full phase is divided into during the COMSOL simulation. It is needed to get correct results from Simulink for the intended machine design. As useful additional information, the mass and volume of the machine can easily be analyzed and exported from COMSOL. This information can then be used when analyzing the Simulink results.

3.2 MATLAB model

MATLAB is a programming and numeric computing platform that can be used for many engineering and computing purposes. In this thesis, MATLAB is used to simulate the time domain behaviour of the TFMS using the data produced by the static COMSOL simulations. Simulink is used to build the block model for the time domain simulation. MATLAB script is used to control the Simulink model and to process the data to be used in Simulink. More information about the commands and blocks used in this model can be found in MATLAB documentation [31].

3.2.1 MATLAB script

The purpose of the MATLAB script is to rearrange the data from the COMSOL simulation for the LUTs and to set all the values to the variables needed for the Simulink model. The basic idea is that there would be no need to change anything from the Simulink model and instead all the control of the simulation can be done from the MATLAB script.

3.2.1.1 Data processing

The data from the COMSOL simulation comes in a list, in a form that is not directly applicable to the LUTs. Therefore, some reshaping of the data needs to be done. The important data are the flux linkage and torque as a function of the rotor angle and the current.

The read data is first scaled in a way that the winding turn number used in the COMSOL simulation does not matter and that the winding turn number in Simulink can be freely changed. It is done by scaling the COMSOL data to correlate the situation with only one winding turn. Then it is possible to scale the data according to the number of winding turns set to Simulink during the simulation. For the scaling to work, the number of winding turns in COMSOL simulation needs to be known. The scaling allows more freedom in choosing the Simulink simulation parameters.

After reading the data from the file containing COMSOL data to MATLAB, all the values of the flux linkage are in one long list. In this form, the data is not usable for the LUTs. Instead, the values need to be arranged to their corresponding positions into a table that has the current values on the y-axis and the rotor angle values on the x-axis. From this kind of table, it is possible to pick the flux linkage value that corresponds to the certain rotor angle and current values.

To save time and to decrease the computational burden, the COMSOL simulations were done only with positive current values. It is possible with reluctance machines, because with positive and negative current values the magnitude of the absolute value of the flux

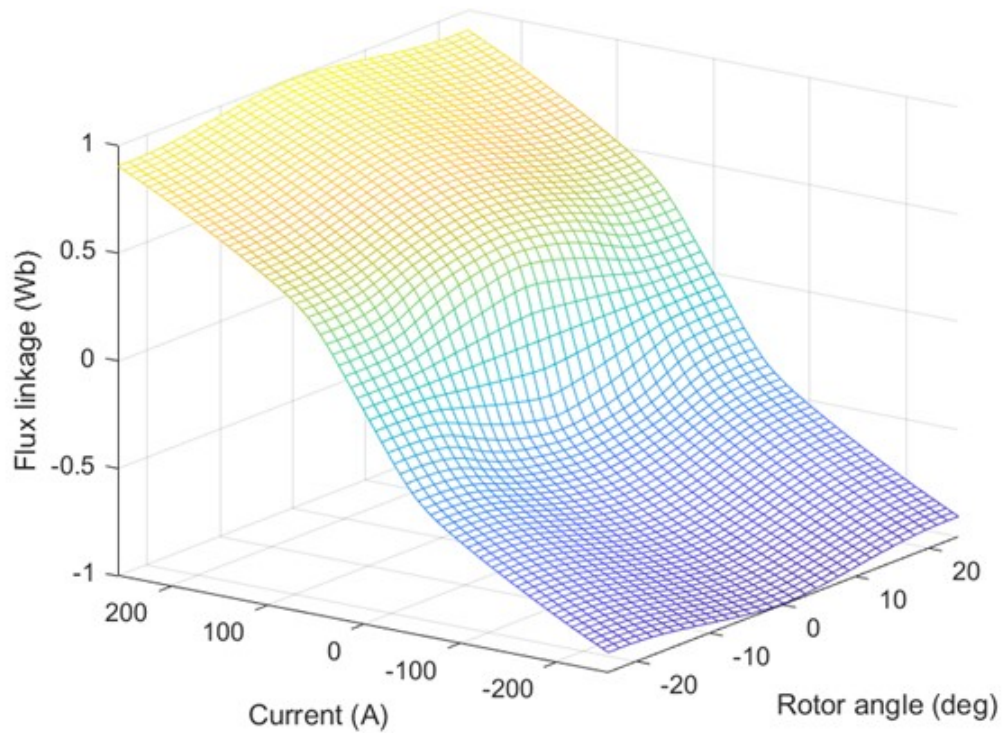


Figure 3.7. An example of the flux linkage data stored to the LUT.

linkage is of the same, but opposite sign. Therefore, the flux linkage values for the negative current values can be obtained by just changing values of the flux linkage with the positive current into negative.

The rearranging and the adding of the negative current values is done to both the flux linkage and torque values to create the LUTs from them both. Examples of the data stored to LUTs as a visual representation can be seen in Figures 3.7 and 3.8.

As it can be seen from Figures 3.7 and 3.8, the flux linkage and the torque value corresponding to the current and the rotor angle can be taken from the LUTs. To ensure the simulation accuracy, it is important that the current in the Simulink simulation does not exceed the maximum values of the LUT. Because of that, the COMSOL simulation should be done with high enough maximum current to be enough for the Simulink simulation. Increasing the number of current values in the COMSOL simulation will increase the number of static simulations needed to be solved, which causes the simulation to take more time. Therefore, estimating the required current level for the Simulink simulation before doing the COMSOL simulation is important.

When solving the current based model shown in (2.4), instead of the flux linkage, the partial derivatives of the flux linkage with respect to the current and with respect to the rotor angle need to be stored to the LUTs. The partial derivatives are obtained by using the *gradient* function in MATLAB. It returns the values for the partial derivatives and they

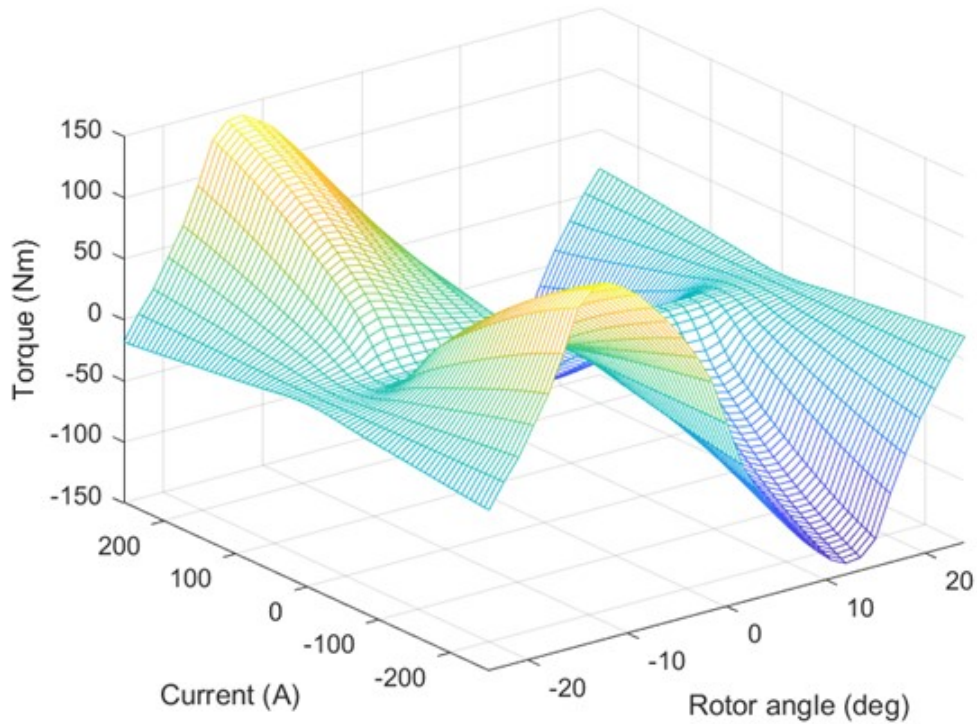


Figure 3.8. An example of the torque data stored to the LUT.

can then be placed to the LUTs. The data of the LUTs containing the partial derivatives are presented in Figures 3.9 and 3.10.

3.2.1.2 Setting the parameters

To run the Simulink model certain parameters need to be set beforehand. This is done in the MATLAB script to avoid the need to manipulate the Simulink model. That will make the simulation easier and faster to control.

The supply line-to-line voltages are

$$\mathbf{U} = \sqrt{2}U_{\text{rms}}\cos(\omega t + \boldsymbol{\theta}) \quad (3.12)$$

where U_{rms} is the RMS value of the voltage, ω is the electric angular frequency, t is the time and $\boldsymbol{\theta}$ contains the phase differences between the line-to-line voltages. The RMS value of voltage is set as a parameter and it is changeable, as long as it does not cause the current to exceed the LUT limitations.

The electrical angular frequency is calculated by

$$\omega = 2\pi f, \quad (3.13)$$

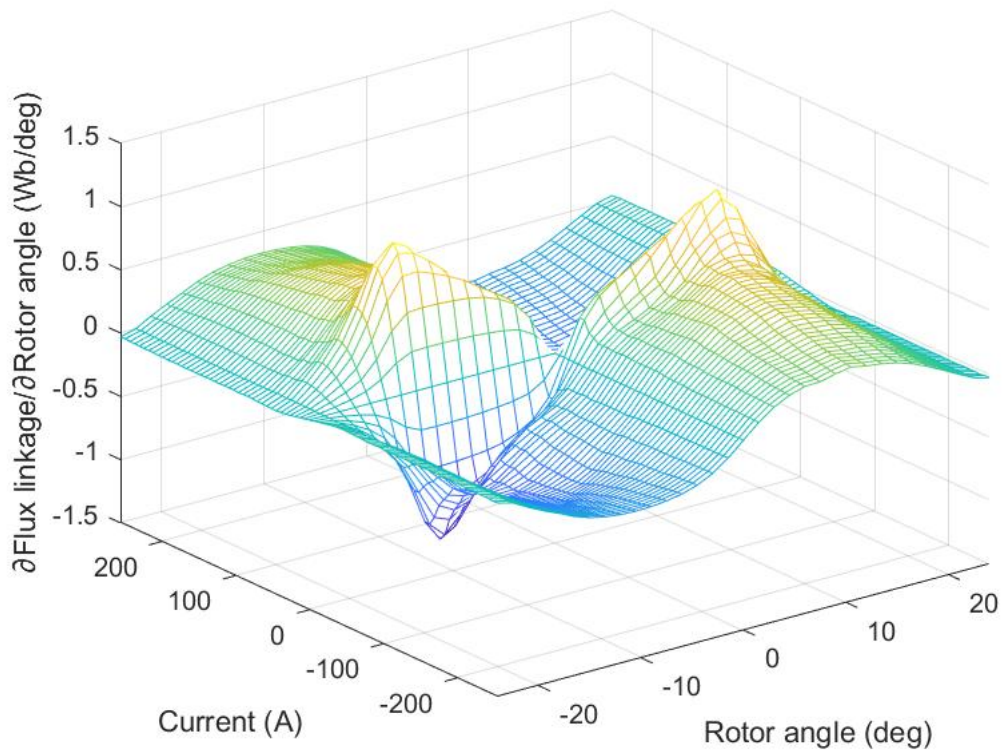


Figure 3.9. An example of the partial derivative of the flux linkage with respect to the rotor angle stored to the LUT.

where f is the frequency of the current waveform, which is set as a parameter and freely changeable.

By taking the chosen number of phases into account, the phase difference can be calculated as

$$\theta = -[0, \dots, m - 1]^T \frac{2\pi}{m}, \quad (3.14)$$

where m is the number of phases. It gives a vector containing the phase differences for each phase. For example, with three phases the phase differences in degrees for the phases are $[0^\circ, -120^\circ, -240^\circ]$ and with five phases the phase differences are $[0^\circ, -72^\circ, -144^\circ, -216^\circ, -288^\circ]$.

The resistance of the stator windings is also a parameter that can be easily changed. It is also possible to set different resistance value for each stator winding, if needed. In all the simulations done for this thesis, the resistance was assumed to be same in every phase.

The rotor shift for each phase depends on the numbers of phases and symmetry sectors. The rotor phase shifts are calculated as

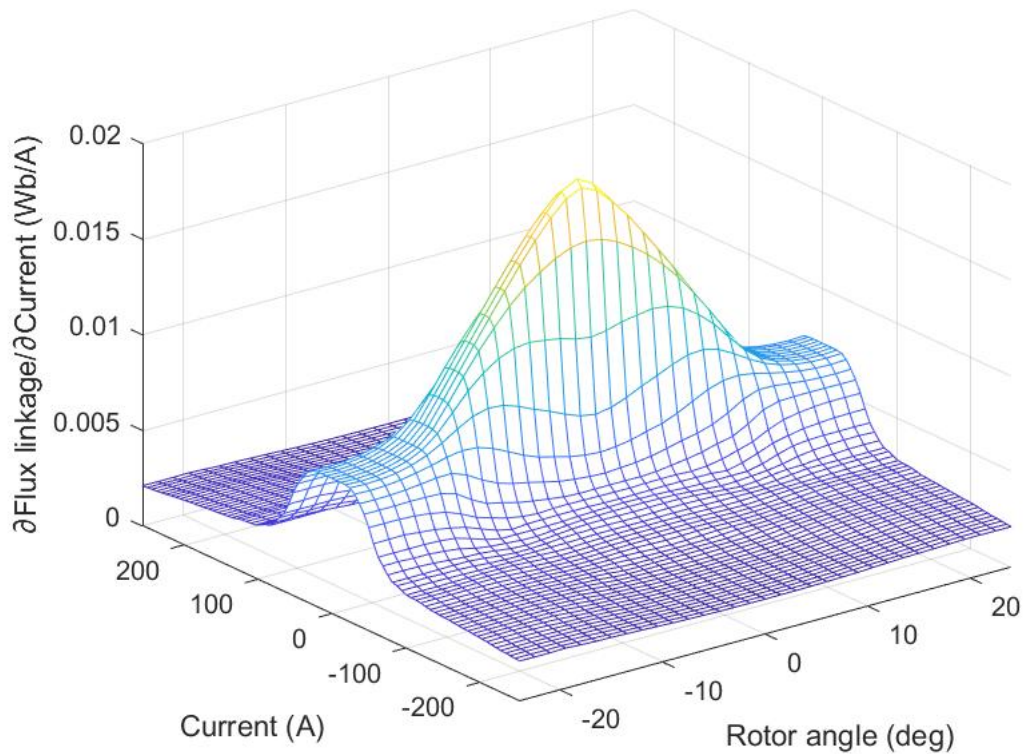


Figure 3.10. An example of the partial derivative of the flux linkage with respect to the current stored to the LUT.

$$\Delta\theta = \begin{bmatrix} 0 \\ \vdots \\ m-1 \end{bmatrix} \frac{2\pi}{mn_{\text{sect}}}, \quad (3.15)$$

where n_{sect} is the number of symmetry sectors. The phase with no rotor shift has the rotor shift of 0° and the rest of the phase shifts depend on the number of phases. For example, with three phases and eight symmetry sectors the rotor shifts in degrees for the phases are $[0^\circ, -15^\circ, -30^\circ]^T$.

The initial rotor angle can be changed by setting the parameter α_0 . It sets the starting position of the rotor and as a default it is set to be 0° . It means that the poles of the stator and rotor of the first phase are directly against each other, as is shown in Figure 2.1. By manipulating the initial rotor angle, the torque produced by the motor can be varied.

The number of phases can be chosen by setting the parameter m . It has an effect on many other variables like the voltage and rotor phase shifts mentioned above and the matrices \mathbf{K} and \mathbf{Q} shown in (2.10) and (2.11).

The simulation works reliably with three or more phases. With one phase the simula-

tion also works reliably. One phase simulation is used in this thesis work to validate the Simulink model by comparing its results to the results of the time domain COMSOL simulation.

The connection between the delta and the star connection can be easily switched by only changing one parameter. When the star connection is chosen, the matrices \mathbf{K} and \mathbf{Q} will be generated as shown in (2.10) and (2.11). When the delta connection is chosen, the matrices \mathbf{K} and \mathbf{Q} will be identity matrices with ones in the main diagonal and zeros elsewhere.

The mechanical angular frequency is calculated by

$$\omega_r = p_c 2\pi f, \quad (3.16)$$

where f is the frequency of the rotation of the rotor, which in this case is the same as the frequency of the supply voltage, and p_c is a parameter, which is used to set the frequency of the current in relation to the rotation speed of the rotor. More about the parameter p_c is explained in the next section.

3.2.2 Relation of current frequency and rotor speed

While developing the Simulink model, it became clear that relation of the rotor speed and frequency of the current needs to be set differently for ideal torque production. At first, the relation was set in a way that the stator and rotor poles were aligned once per each supply period. This method did not give as good torque production as expected.

With reluctance motor, the direction of the current does not matter, the rotor pole will try to lock on to stator pole with both positive and negative currents. Therefore, much better results were obtained when the frequency of the supply current was chosen so that the stator and rotor poles were aligned twice per supply period.

This was done in the Simulink model with a variable p_c . It is used to slow down the frequency of the current compared to the rotational frequency of the rotor. When the value of p_c is 2, the poles align twice per supply period and the torque production is improved. Below is a proof that the p_c value should be 2.

The phase inductance can be written as a sum of average and fundamental components as

$$L_j = L_{av} + \Delta L \cos(n_{sym}\alpha + \Delta\theta_j), \quad (3.17)$$

where n_{sym} is the number of symmetry sectors, α is the position angle of the rotor and $\Delta\theta_j$

is the rotor shift of phase j . In this case, there are as many symmetry sectors as there are rotor poles, so the number of symmetry sectors is used in these calculations instead of the number of rotor poles. When the inductance is differentiated with the respect to α , we get

$$\frac{dL_j}{d\alpha} = -n_{\text{sym}}\Delta L \sin(n_{\text{sym}}\alpha + \Delta\theta_j). \quad (3.18)$$

Phase j current, which is slowed down by the factor p_c compared to the rotor speed, is

$$i_j = \cos\left(\frac{n_{\text{sym}}\alpha + \Delta\theta_j}{p_c} + \phi\right), \quad (3.19)$$

where ϕ is the angle between the current and the voltage vectors.

The torque of phase j is

$$T_j = \frac{1}{2} \frac{dL_j}{d\alpha} i_j^2. \quad (3.20)$$

When equations (3.18) and (3.19) are inserted to the torque equation, it takes the form

$$T_j = \frac{1}{2} \left(-n_{\text{sym}}\Delta L \sin(n_{\text{sym}}\alpha + \Delta\theta_j) \cos^2\left(\frac{n_{\text{sym}}\alpha + \Delta\theta_j}{p_c} + \phi\right) \right). \quad (3.21)$$

Equation (3.21) can be simplified by using the trigonometric identity

$$\cos(x)^2 = \frac{1 + \cos(2x)}{2}, \quad (3.22)$$

and the torque becomes

$$T_j = \frac{1}{2} \left(-n_{\text{sym}}\Delta L \sin(n_{\text{sym}}\alpha + \Delta\theta_j) \frac{1}{2} \left(1 + \cos\left(2\frac{n_{\text{sym}}\alpha + \Delta\theta_j}{p_c} + \phi\right) \right) \right). \quad (3.23)$$

By completing the multiplication, the equation is

$$T_j = \frac{-n_{\text{sym}}\Delta L}{4} \left(\sin(n_{\text{sym}}\alpha + \Delta\theta_j) + \sin(n_{\text{sym}}\alpha + \Delta\theta_j) \cos\left(\frac{2}{p_c}(n_{\text{sym}}\alpha + \Delta\theta_j) + \phi\right) \right). \quad (3.24)$$

The resulting equation can be further simplified by using the trigonometric identity

$$\sin(\alpha)\cos(\beta) = \frac{1}{2} \left(\sin(\alpha + \beta) + \sin(\alpha - \beta) \right), \quad (3.25)$$

resulting in

$$T_j = \frac{-n_{\text{sym}}\Delta L}{4} \left(\sin(n_{\text{sym}}\alpha + \Delta\theta_j) + \frac{1}{2} \sin \left(n_{\text{sym}}\alpha + \Delta\theta_j + \left(\frac{2}{p_c}(n_{\text{sym}}\alpha + \Delta\theta_j) + \phi \right) \right) \right. \\ \left. + \frac{1}{2} \sin \left(n_{\text{sym}}\alpha + \Delta\theta_j - \left(\frac{2}{p_c}(n_{\text{sym}}\alpha + \Delta\theta_j) + \phi \right) \right) \right). \quad (3.26)$$

With some rearranging, the equation can be expressed as

$$T_j = \frac{-n_{\text{sym}}\Delta L}{4} \left(\sin(n_{\text{sym}}\alpha + \Delta\theta_j) + \frac{1}{2} \sin \left((n_{\text{sym}}\alpha + \Delta\theta_j) \left(1 + \frac{2}{p_c} \right) + \phi \right) \right. \\ \left. + \frac{1}{2} \sin \left((n_{\text{sym}}\alpha + \Delta\theta_j) \left(1 - \frac{2}{p_c} \right) - \phi \right) \right). \quad (3.27)$$

From (3.27), it can be seen that both

$$\sin(n_{\text{sym}}\alpha + \Delta\theta_j) \quad (3.28)$$

and

$$\frac{1}{2} \sin \left((n_{\text{sym}}\alpha + \Delta\theta_j) \left(1 + \frac{2}{p_c} \right) + \phi \right) \quad (3.29)$$

have average of zero over $0 \leq \alpha \leq 2\pi$, but

$$\frac{1}{2} \sin \left((n_{\text{sym}}\alpha + \Delta\theta_j) \left(1 - \frac{2}{p_c} \right) - \phi \right) \quad (3.30)$$

is nonzero over $0 \leq \alpha \leq 2\pi$, if $p_c = 2$. Therefore, the motor produces torque only when the value of p_c is 2.

3.2.3 Simulink model

Simulink is a graphical programming environment in MATLAB. It can be used for modelling and simulating dynamic systems. The models are built using blocks from different block libraries. The blocks can be, for example, integrators, constants or many other mathematical operators. The Simulink model of TFMS built for this thesis can be seen in Appendix A.

The model shown in Appendix A solves the voltage equation (2.7). The inputs, that can be given to the model, are the line-to-line voltages and the mechanical angular frequency. The matrices K and Q are modified in the script depending on the phase number, as they were explained earlier with the winding connection. The LUTs are constructed from the 3D FEM simulation data processed in the script.

3.2.3.1 Input values

The input values that can be easily altered are the line-to-line voltages and the mechanical angular frequency. These parameters are used to control the simulation and they are changed depending on the need.

A ramp function is used to limit the start values of the voltages. The ramp function is set to reach the saturation point at $1/(f/2)$, where f is the frequency of the supply voltage. After the saturation point, the ramp function has no effect on the simulation. The start values of the voltages are limited to avoid major initial transients at the beginning of the simulation. It allows the simulation to reach the steady state operation a bit faster.

The mechanical angular frequency is provided to calculate the angle of the rotor. It can be set in the script to alter the behaviour of the simulation. The rotor phase shift depends on the number of phases. When the angle of the rotor is calculated, it is added with the rotor phase shifts to get the rotor angles of each phase. They can then be used when picking the required values from the LUTs.

3.2.3.2 Lookup tables

To solve the voltage equation (2.7) in Simulink, LUTs for the partial derivatives of the flux linkage with respect to both the current and the rotor angle are needed. To get the torque of the machine, a LUT for torque is needed. The subsystem containing the LUT operations is depicted in Figure 3.11.

Lookup Table (n-D) block is used to create the LUTs using the data processed with the script. The input values to the subsystem are the current and the rotor angle vectors. They are used to search the corresponding values of the torque and the partial derivatives for each phase from the LUTs. The torque values for each phase are then added together to get the torque of the whole machine.

The range of the LUTs is limited by the data stored in them. The only way to increase the range is to increase the number of COMSOL simulations, which increases the computational burden and it takes more time to compute. Therefore, it is useful to roughly predict the needed maximum current when running the COMSOL simulation. If the LUT receives an out-of-range input, it will cause an error message and end the simulation.

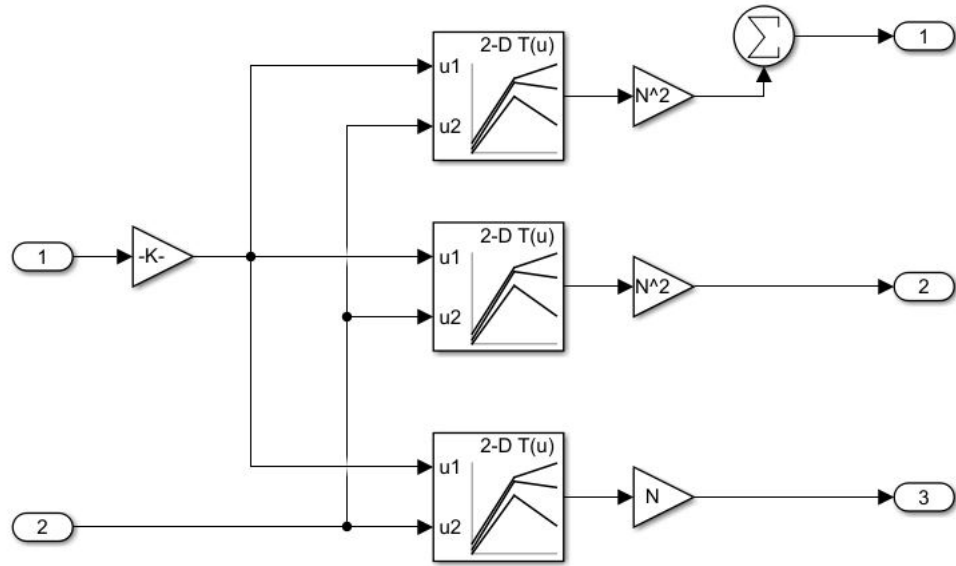


Figure 3.11. The subsystem containing the LUTs.

3.2.3.3 Simulation outputs

Some of the data needs further calculating in addition of solving the voltage equation. This can be done by utilizing the already known variables from the model. By using the Simulink blocks, these variables can be used in calculations.

An example of a variable that can be used in further calculations is the produced torque of the machine. Torques produced by each phase are available from the LUT when the current is solved and the rotor angle is known. Torques generated by each phase can then be added up together to get the torque of the full machine. Then the time average of the torque is easily calculated by using the *Mean* block in Simulink.

For example, by knowing the produced torque, the mechanical power P_{mech} of the machine can be calculated as

$$P_{\text{mech}} = T\omega_r, \quad (3.31)$$

where T is the torque and ω_r is the mechanical angular velocity of the rotor. The time average of the mechanical power can then be calculated using the *Mean* block. Then as an output the mean of the mechanical power in every time step can be transferred to the workspace.

An example of useful information to know are the phase voltages. The input voltages are not the same because they are line-to-line voltages. When using the star connection, the line-to-line voltages and the phase voltages are not equal. Therefore, the phase voltages are calculated by using the voltage equation in the form shown in (2.3). This equation

can be built using Simulink blocks. The resulting phase voltages can then be used when solving other desired simulation outputs like electric power.

3.2.4 Iron core losses

In addition to the basic model of the machine, iron core loss calculations are also added to the Simulink model. LUTs are also used for the loss calculation model to store the data of magnetic flux density distribution, which is obtained from the series of static simulations of the FE model. This way to implement the loss calculation allows fast estimation of the iron core losses in synchronous machines. A model of an outer rotor transversal flux permanent magnet machine, developed by VTT, is used to demonstrate the operation of the loss calculation to get more realistic results.

The loss calculation method could be generalized to consider each component of the flux density vector, but in this model only the norm of the flux density data

$$B_{\text{FE}}(\mathbf{r}, i, \alpha_r) = \sqrt{B_{\text{FE},x}^2(\mathbf{r}, i, \alpha_r) + B_{\text{FE},y}^2(\mathbf{r}, i, \alpha_r) + B_{\text{FE},z}^2(\mathbf{r}, i, \alpha_r)} \quad (3.32)$$

is considered for simplicity. $B_{\text{FE}}(\mathbf{r}, i, \alpha_r)$ is numerically represented by the assembly of flux density values in the integration points of the FE mesh, denoted by the column vector $\mathbf{b}_{\text{FE}}(i, \alpha_r)$. A visual illustration of the integration points in a FE mesh of one symmetry sector of the machine developed by VTT is shown in Figure 3.12.

The red dots in Figure 3.12 are the integration points of the stator core and the black dots are the integration points of the rotor core. In this example, the number of integration points in the stator is 3331 and in the rotor 1801. The number of integration points depends entirely on the geometry of the machine and on the element size of the FE mesh.

The density of integration points is high, for example, at the ends of the stator poles, because the FE mesh is defined to be finer in those important areas. Because of the norm of the flux density data is used, there is one value of flux density for each integration point for each combination of current and rotor angle. If every component of the flux density vector is considered separately, instead of considering only the norm, the amount of data needed to be stored increases.

Once the solution for the stator currents $i_j(t)$ ($j = 1, \dots, m$) is obtained from (2.7) and rotor position $\alpha(t)$ is known, the time-variation of the flux density distribution in the core of phase j can be evaluated as

$$\mathbf{b}_j(t) = \mathbf{b}_{\text{FE}}(i_j(t), \alpha(t) - \Delta\theta_j). \quad (3.33)$$

When the time variation of the flux density distribution is known, a suitable iron loss model

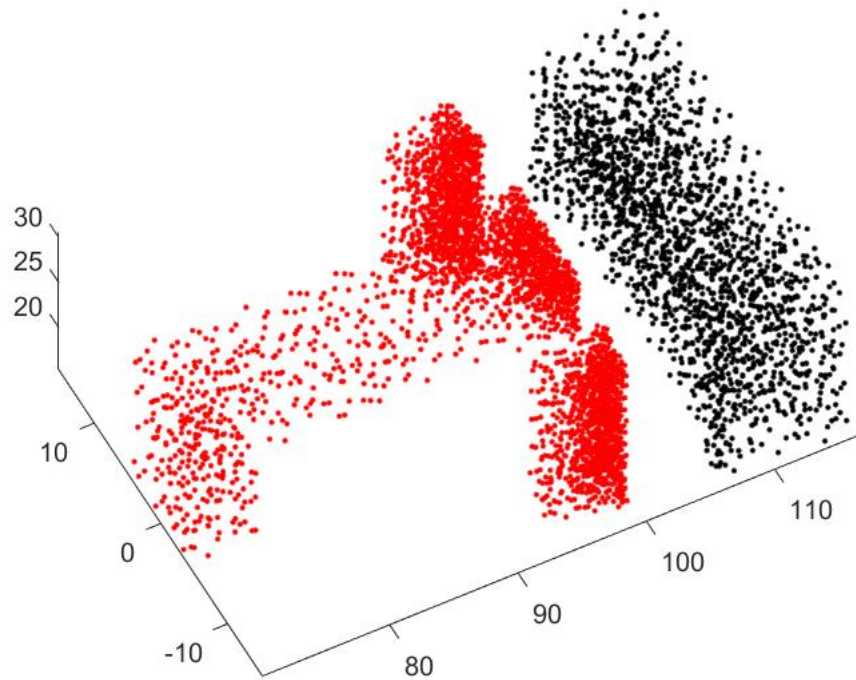


Figure 3.12. An example of integration points of one symmetry section of the machine.

can then be applied to estimate either the instantaneous or time averaged power density in the core. For simplicity, we denote the instantaneous power density obtained from such a model as

$$\mathbf{p}_{\text{FE},j}(t) = p(\mathbf{b}_j(t)), \quad (3.34)$$

denoting that scalar function p is applied to each element of $\mathbf{b}_j(t)$ separately. The total core loss $P_{\text{FE},j}(t)$ can then be obtained by integrating over the core volume, i.e., evaluating the sum

$$P_{\text{FE},j}(t) = \mathbf{w}^T \mathbf{p}_{\text{FE},j}(t) = \mathbf{w}^T p(\mathbf{b}_j(t)), \quad (3.35)$$

where \mathbf{w}^T is a row vector containing the integration weights.

A 3D FE mesh for one symmetry sector of a TFMS may contain tens of thousands of integration points, which makes the evaluation of iron core losses prohibitively time-consuming. A computationally lighted method is thus needed to approximate the core losses.

3.2.5 Proper orthogonal decomposition

To make the simulations faster by reducing the size of the solved system, proper orthogonal decomposition (POD) [32] is used. It is a numerical method that can be used to reduce the size of computationally heavy simulations. When using POD, a reduced basis is formed from data that needs to be reduced. The original not reduced data can be called an original basis. A discrete projection operator can then be determined to get an estimation of the original basis from the reduced basis. In this model POD is used to reduce the amount of data stored to the LUTs. For that reason, a reduced basis needs to be made from the flux density distribution data.

The so called snapshot matrix \mathbf{S} can be formed by arranging the flux density values obtained from the FE simulations for the n_i current and n_α rotor position values to $n_i n_\alpha$ columns as

$$\mathbf{S} = \left[\mathbf{b}_{\text{FE}}(i_1, \alpha_1) \quad \mathbf{b}_{\text{FE}}(i_1, \alpha_2) \quad \cdots \quad \mathbf{b}_{\text{FE}}(i_{n_i}, \alpha_{n_\alpha}) \right]. \quad (3.36)$$

In the example case, the dimension of the snapshot matrix for the stator is 3331×861 and for the rotor it is 1801×861 . Basically, the number of columns is the number of different combinations of the current and rotor angle values and the number of rows is the number of integration points in the stator or rotor mesh.

The POD basis is then searched by applying the singular value decomposition to \mathbf{S} in the standard manner. It is done by using existing MATLAB function *svd*. This gives the POD mapping \mathbf{P} from the reduced basis to the full basis. By the pseudo-inverse of \mathbf{P} , we can express the magnetic flux density distribution in a reduced basis

$$\mathbf{b}_{\text{FE}}^*(i, \alpha) = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T \mathbf{b}_{\text{FE}}(i, \alpha), \quad (3.37)$$

which can potentially reduce the size from tens of thousands to only a few. This can reduce the amount of data stored to the LUTs significantly, because only the reduced basis is needed to be stored to the LUTs. The set of values corresponding to the current and rotor angle values can then be picked from the LUT containing the reduced flux density values. These values can then be mapped back to the actual flux density values for the loss calculations. The original basis can be estimated by

$$\mathbf{b}_{\text{FE}}(i, \alpha) = \mathbf{P} \mathbf{b}_{\text{FE}}^*(i, \alpha). \quad (3.38)$$

When forming the POD basis, a tolerance for the POD forming needs to be defined. This tolerance determines the achieved amount of data reduction. With a very tight tolerance,

the data is not as reduced as with less tight tolerance. Tighter tolerance, for the POD basis creation, decreases the amount of error when approximating the original basis from the reduced basis. However, the increased accuracy comes with a cost of requiring more data to be stored.

Using POD can make the simulations significantly faster, but also increases the possibility for error in the simulation results. For this reason, it is important to find a good balance between the simulation speed and the result accuracy and adjust the POD forming in an appropriate way. Examples of how the simulation times and results can be affected by changing the tolerance for the POD forming are shown later in the Results chapter.

3.2.6 Homogeneous regions

In addition to POD, the computational burden can be decreased by identifying regions from the FE mesh, in which the flux-density distribution is approximately homogeneous independently of the values of the current i and the rotor angle α . This can be done by studying the rows of P and S . Basically, it means that integration points where the flux density values are very similar can be handled as a group instead of handling every integration point independently. An example of integration points, in the stator of the machine developed by VTT, where the flux density distribution can be considered to be approximately homogeneous is shown in Figure 3.13.

The black dots in Figure 3.13 are the integration points of the stator. The points that are marked by the red circles can be considered to have approximately homogeneous flux density distribution and thus be considered as one region. The number of the regions depends on the uniformity of the flux density distribution and on the tolerance that is used when creating the regions.

As is with the POD forming, also the homogeneous region forming needs to have some tolerances determined. With a very tight tolerance, the flux density values in the integration points need to be very close to each other to be considered as a part of the same homogeneous region. With less tight tolerance, the range of the flux density values considered to be part of the same region is bigger.

This tolerance will also have an effect on the speed and accuracy of the simulation. With a tight tolerance, more data needs to be stored, but the accuracy of the simulation is better than with less tight tolerance. Similarly to the POD forming, a good balance between the speed and accuracy of the simulation needs to be found when the tolerances are determined.

It is then possible to express the magnetic flux density distribution $\mathbf{b}_{FE}(i, \alpha)$ as

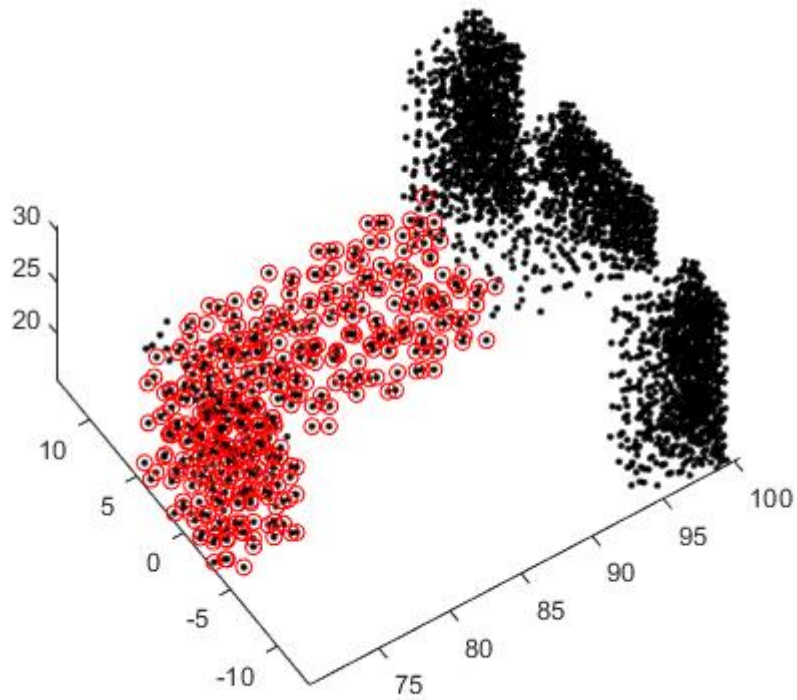


Figure 3.13. An example of integration points with approximately homogeneous flux density distribution.

$$\mathbf{b}_{\text{FE}}(i, \alpha) = \mathbf{M}\mathbf{b}_{\text{FE}}^{\text{h}}(i, \alpha), \quad (3.39)$$

where $\mathbf{b}_{\text{FE}}^{\text{h}}(i, \alpha)$ contains the flux-density values of each homogeneous region and \mathbf{M} is a selection matrix, which maps the flux density values from the homogeneous regions to the full distribution. We can thus apply the iron loss model only to the flux density values $\mathbf{b}_j^{\text{h}}(t)$ evaluated in the homogeneous regions and integrate over the volume as $\mathbf{w}^{\text{hT}}\mathbf{M}p(\mathbf{b}_j^{\text{h}}(t))$, where \mathbf{w}^{hT} contains the volumes of the regions. This can significantly lower the computational burden for calculating the iron core losses.

4. RESULTS

In this chapter, some results of the simulations are presented. To obtain the iron core loss simulation results, the permanent magnet transversal flux machine developed by VTT was used. The Simulink model validation and other results are obtained by using the simple model of TFRM modelled for this thesis.

4.1 Validating the Simulink model

Validating the Simulink model is done by comparing the results from the Simulink simulation and the results from the time domain simulation from COMSOL. Same settings were used for both simulation methods to get comparable results. Used settings are shown in Table 4.1.

U_{rms}	100 V
N	100
f	50 Hz
R_s	1.031 Ω
m	1
α_0	0°

Table 4.1. Parameter settings used for the comparison simulations.

Parameter U_{rms} is the RMS value of the voltage, N is the number of stator winding turns and f is the frequency of the supply voltage. R_s is the stator winding resistance and it is the value determined by COMSOL. m is the phase number and only one phase was used because the COMSOL simulation only solves one phase. α_0 is the initial value of the rotor angle, which was chosen to be 0° for simplicity.

All the needed variables were calculated in the same way in both simulation methods. The simulation time was set in a way that the steady state operation was achieved and sustained for a while. Then it is possible to compare both the initial transients and the steady state operation. The initial transients of current and torque waveforms with both Simulink and COMSOL are depicted in Figure 4.1 and the steady state is depicted in

Figure 4.2.

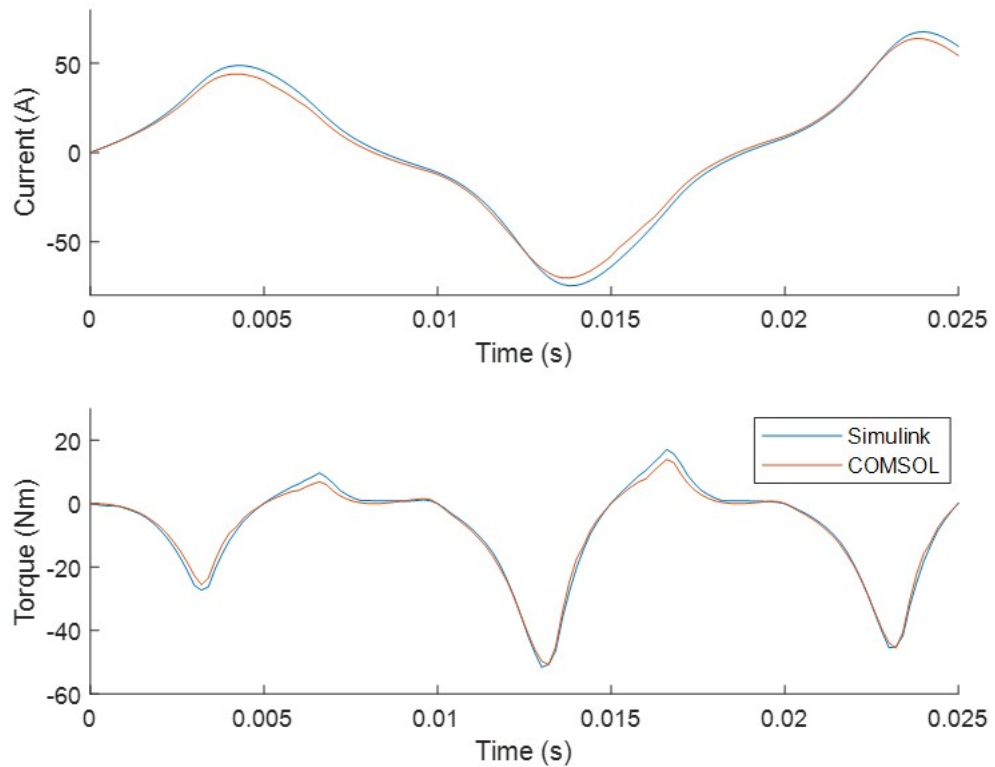


Figure 4.1. Current and torque initial transient with Simulink and COMSOL time domain simulations.

As seen from Figures 4.1 and 4.2, both transient and steady state seem to match quite well when comparing the results from Simulink and COMSOL. The peaks of the waveforms seem to be bit higher with the Simulink simulation, but overall the waveforms match very well. Even the transient at the beginning of the simulation seems to be very similar with both methods.

The results of both simulation methods are very similar, but the simulation times of different methods are very different. The time domain simulation in COMSOL took multiple hours to finish while the time domain simulation in Simulink finished in seconds. This further proves that it is useful to simulate the time domain simulations with Simulink instead of with COMSOL.

Based on the comparison between the two simulation methods, it can be presumed that the time domain Simulink simulations with the LUT data from the static 3D FEM simulations work as well as the time domain 3D FEM simulations. Next, more results from the Simulink model can be shown to demonstrate more of the features of the Simulink model.

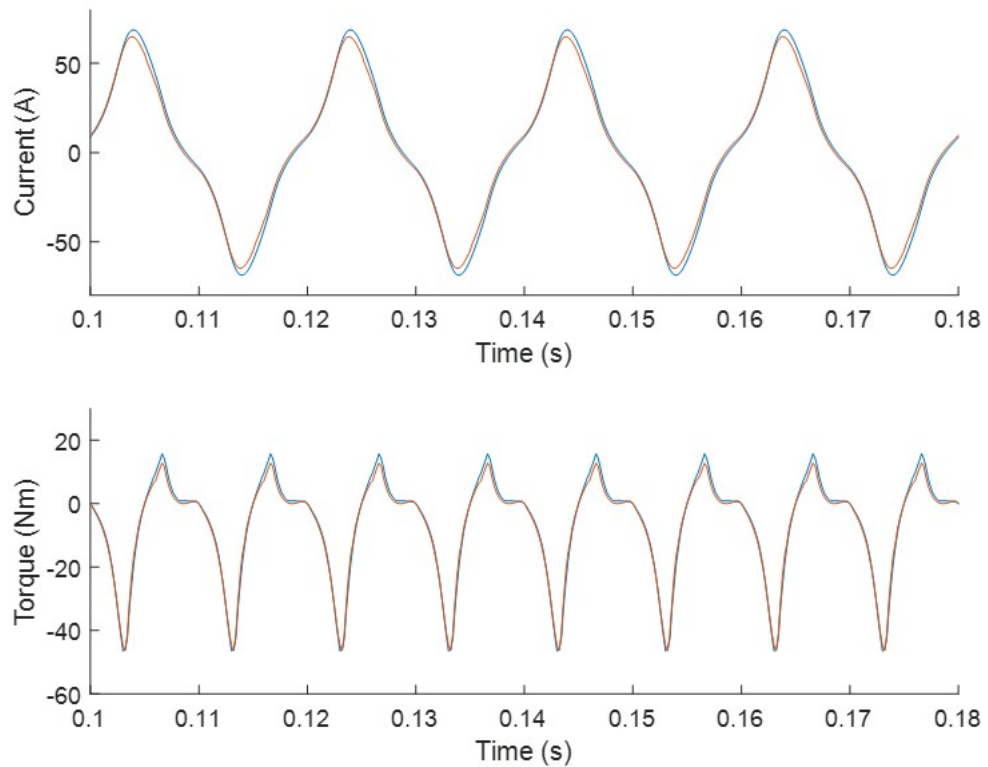


Figure 4.2. Current and torque in steady state operation with Simulink and COMSOL time domain simulations.

4.2 Effects of phase number and connection type

As mentioned above, each of the number of phases and the connection type between star and delta connections can be chosen by only changing one parameter. The effects of changing these settings are demonstrated here. Naturally, the simulation results depend also on the other settings in COMSOL and Simulink simulations, but here only the effects of changing the phase number and the connection type are considered. The other settings are chosen in a way that the Simulink model can be easily demonstrated, instead of trying to demonstrate a realistic machine.

The settings used to demonstrate the machine with different numbers of phases are shown in Table 4.2.

connection	star
U_{rms}	100 V
N	100
f	50 Hz
R_s	0.01 Ω

Table 4.2. Parameter values used to demonstrate the different number of phases.

The settings shown in Table 4.2 are not changed between simulations. Only the number of phases is changed when the effect of phase number is demonstrated. The results of the simulations are shown in Figure 4.3.

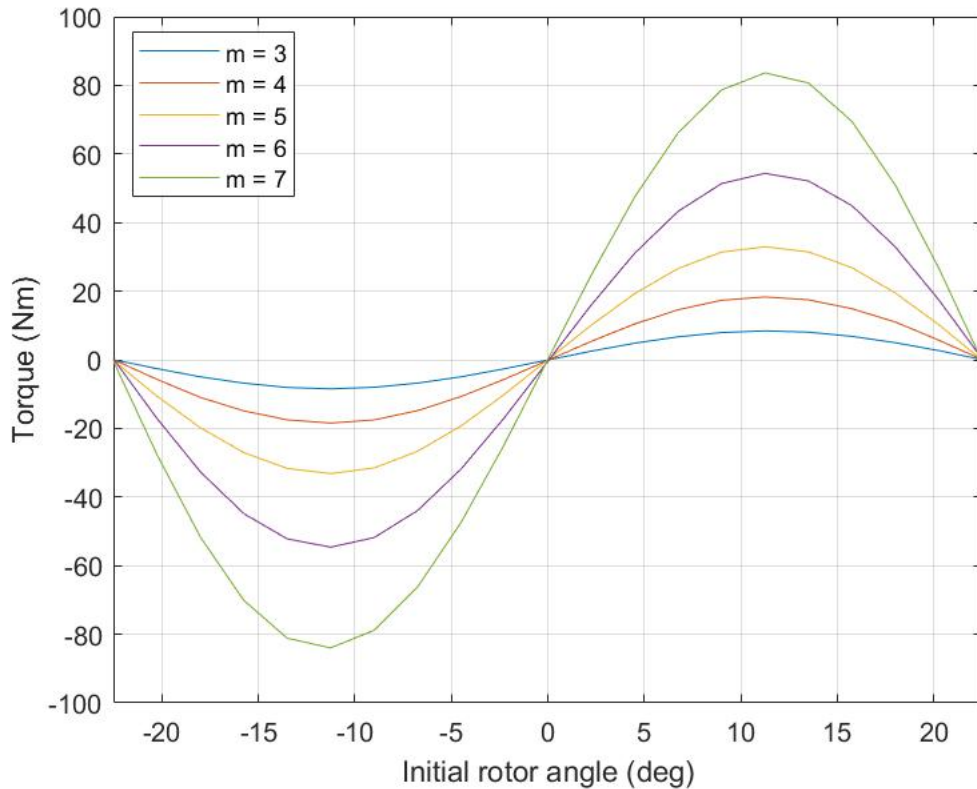


Figure 4.3. Torque as a function of initial rotor angle obtained with different number of phases.

Figure 4.2 shows the torque as a function of the initial rotor angle. Results from five sets of simulations are shown. Each simulation set was done with a different number of machine phases and the number of phases is marked with the parameter m . From the torque curves, it can be seen that the maximum torque increases when the number of phases increase.

One feature of the Simulink model is the easy switching between using the star or delta connection as the winding connection. An example of what kind of effect the winding connection can have to the torque waveform produced by the simulation is presented here. The settings that were kept the same with both connection types are shown in Table 4.3.

The RMS value of the line-to-line voltage U_{rms} was chosen to be 100 V for the delta connection and $100\sqrt{3}$ V for star connection. This is done to get the same phase voltage

m	3
N	100
f	50 Hz
R_s	0.01 Ω

Table 4.3. Parameter values that were kept the same with both connection types.

with both connection types. In this way, the maximum torque produced by the machine with both connection types is approximately the same.

The initial rotor angle was chosen in a way that the produced average torque T_{av} with both connection types was the same. The value for the average torque was chosen as

$$T_{av} = \frac{T_{max}}{1.3}, \quad (4.1)$$

where T_{max} is the maximum average torque produced by the machine. With the simulation settings used here, the maximum average torque produced by the machine was 22.6 Nm. This means that the average torque to be obtained with the simulations was calculated to be 17.4 Nm. This average torque was obtained with the initial torque angle of 6.8° with the star connection and 5.1° with the delta connection. The torque waveforms in steady state operation with both connection types, when the obtained average torque is the same, are shown in Figure 4.4.

As seen in Figure 4.4, there is quite a lot of torque ripple with both connection types, which can be caused by the not very ideal machine design. However, the connection type also seems to have an effect on the torque ripple. It can be seen that with the star connection there is more torque ripple than with the delta connection.

4.3 Iron core losses

To demonstrate the iron core loss calculations, the permanent magnet transversal flux machine developed by VTT is used. That way the losses are slightly more realistic and it was possible to test the Simulink model with different machine design. The torque as a function of initial rotor angle produced by the VTT machine with different numbers of phases is shown in Figure 4.5.

The accuracy of the estimated iron core losses depends on the settings used for the flux density distribution data reduction methods. When estimating the original flux density distribution from the flux density distribution reduced by the POD and from the flux density distribution of the homogeneous regions, there will be some inaccuracy in the estimations when compared to the original values.

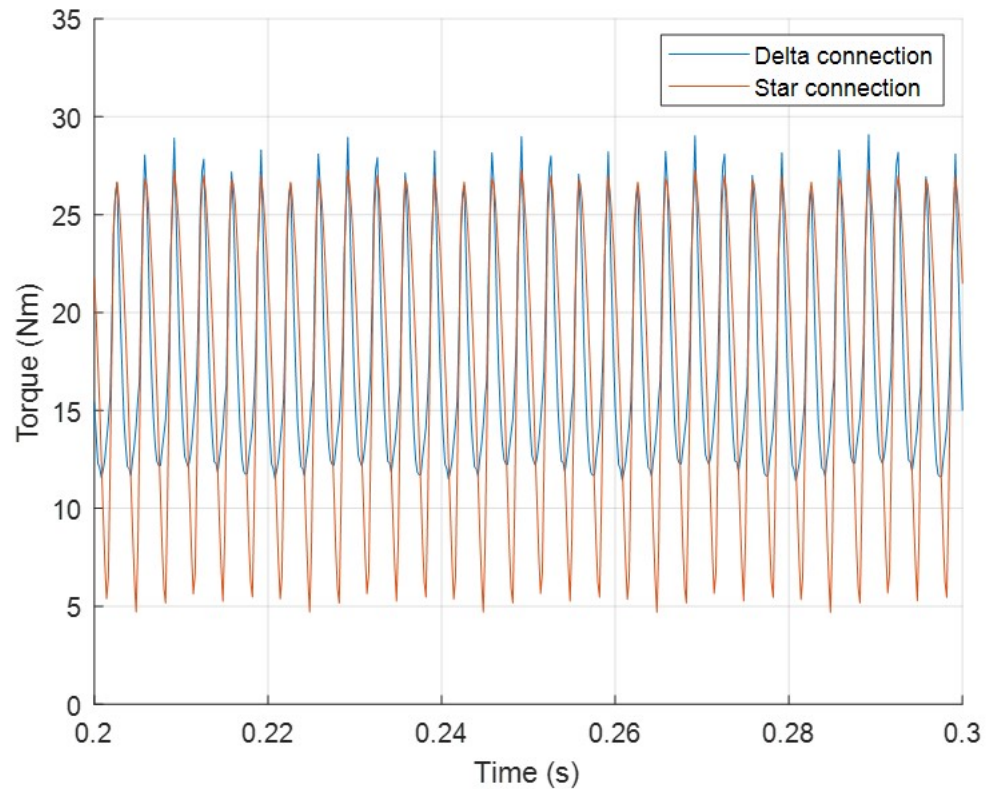


Figure 4.4. The torque waveforms when the obtained average torque is the same with both connection types.

The inaccuracy in the flux density distribution then affects the iron core loss estimation accuracy. For this reason, the inaccuracy of the flux density distribution in an example case is demonstrated here. Figures 4.6 and 4.7 depict the error in the norm of the magnetic flux density distribution in stator and in rotor in the example cases.

The first percentage number above Figures 4.6 and 4.7 describes the error in the average magnetic flux density distribution when only POD is used to reduce the data. The second percentage number describes the error when POD is used and the homogeneous regions are identified. Errors are obtained by comparing the flux density estimations from the reduced flux density values to the original flux density values.

As seen in Figures 4.6 and 4.7, reducing the data by using POD and identifying the homogeneous regions cause some inaccuracy to the magnetic flux density values. This error can be altered by adjusting the tolerances of the POD forming and the homogeneous region identifying, as explained when the reduction methods were discussed.

Adjusting the tolerances will have an influence on the accuracy of the flux density distribution estimation, which in turn affects the accuracy of the iron core losses calculated by utilizing the flux density distribution. The tolerances also have an influence on the simulation time, because the simulation takes longer to complete when there is more data used

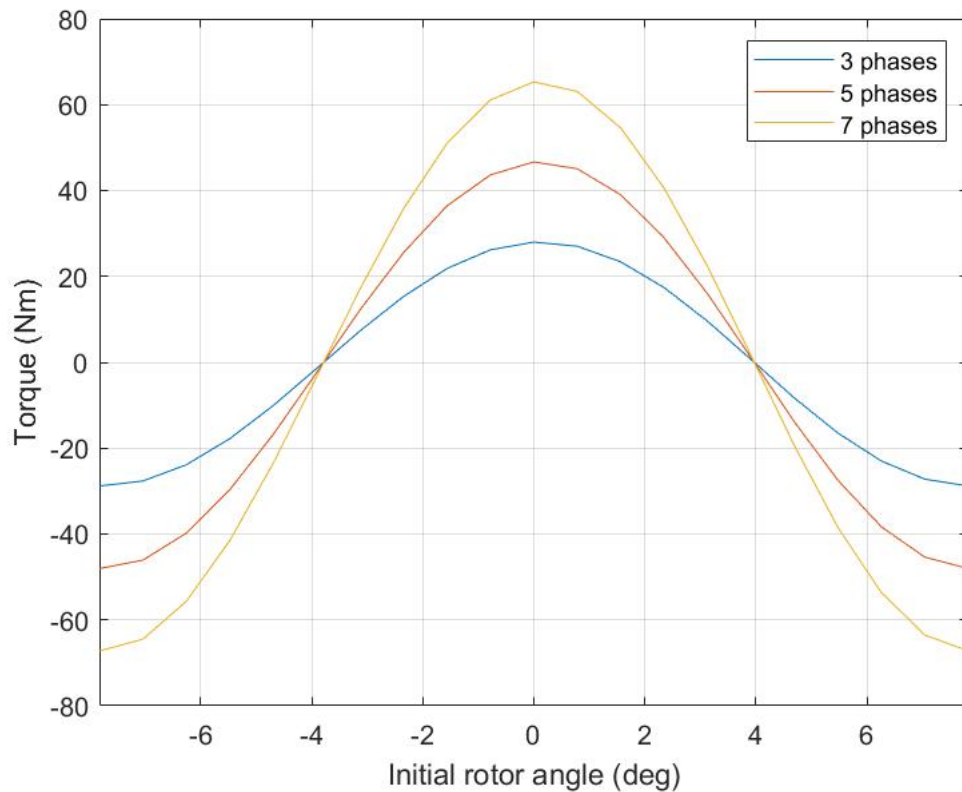


Figure 4.5. Electromagnetic torque as a function of initial rotor angle with 3, 5 and 7 phases.

for the calculations. The effects of the tolerances to the simulation accuracy and simulation time are shown by comparing different example cases. Stator losses as a function of the initial rotor angle are described in Figure 4.8. Table 4.4 contains the simulation times and the sizes of the data.

Only iron core losses of the stator are shown here, because the iron core losses of the rotor were noticed to be quite small for the machine used as an example. For each case with different data reduction settings, 21 simulations with different values for the initial rotor angle were solved.

In the full data case, no POD is used and no homogeneous regions are identified. The simulations are completed using the whole flux density distribution data. The reduced data 1 is a heavily reduced case where POD is used and homogeneous regions are identified. It is also the same case that was used for creating Figures 4.6 and 4.7. In the reduced data 2 case, the tolerance for the POD forming was set to be tighter. In the case of reduced data 3, the POD forming and the homogeneous region identification had tighter tolerances.

As seen in Figure 4.8, the methods to reduce the data do cause some inaccuracy to the calculated stator losses. The curve for the full data case gives the baseline to compare

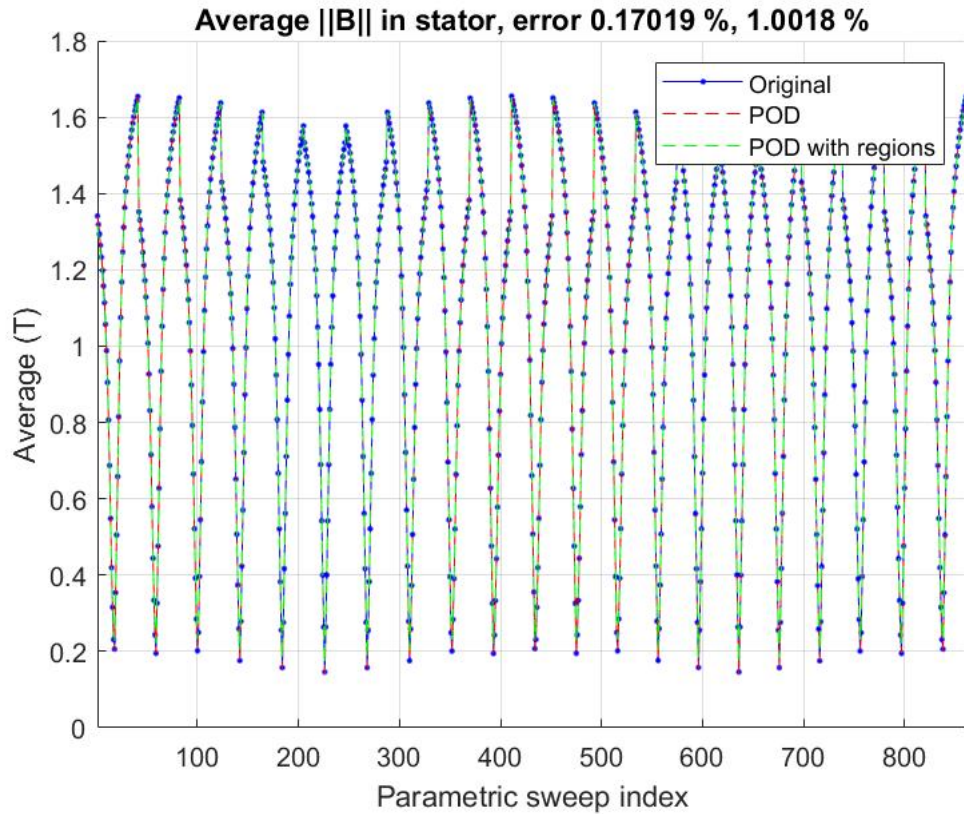


Figure 4.6. An example of the error of the average $\|B\|$ in stator.

the effects of the data reduction. The curves in Figure 4.8 show that the more the data is reduced the more inaccuracy the results will have. When the tolerances are tightened, the calculated losses get closer to the case where no data reduction is used.

From Table 4.4, it can be seen how the data reduction affects the simulation time and the size of the data. Average simulation time is the time that one simulation takes to complete in average. Total simulation time is the time to complete all 21 simulations in the set of simulations. Number of rows in flux density distribution is the number of different rows of flux density values in the matrix containing the data. Number of homogeneous regions describes how many homogeneous regions are identified.

The simulation time is considerably shorter when using the reduced data compared to the simulation time of the full data case. As expected, the simulation time increases when the tolerances of POD forming and homogeneous region identification are tightened. However, even with tighter tolerances the simulation time is much shorter than in the full data case. When compared to the 3D FEM simulations with COMSOL, even the full data case simulation is very fast. One simulation with COMSOL can take over an hour.

The data size is expressed with the number of rows in the flux density distribution and with the number of homogeneous regions. In the full data case, there is a row of flux density values for each integration point. In this example, there are 3331 integration points in

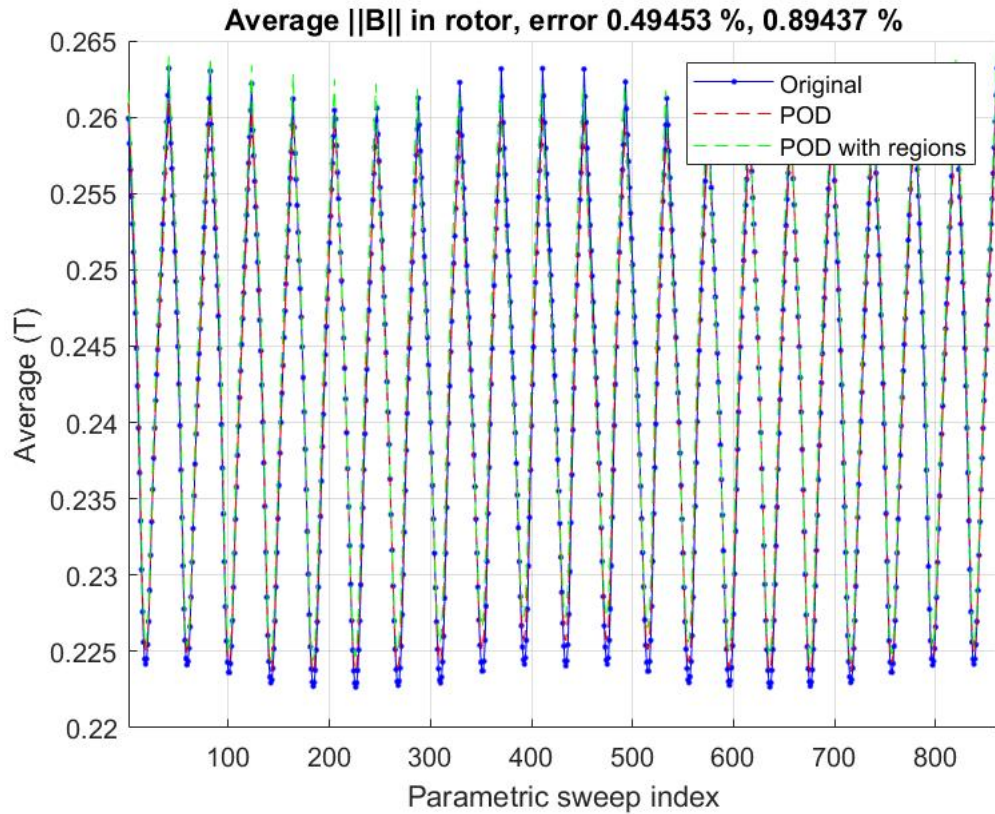


Figure 4.7. An example of the error of the average $\|B\|$ in rotor.

the stator. One of these rows contain the flux density values for each current and rotor angle combination in that integration point. In this example case, the size of the matrix containing the flux density data would be 861×3331 . By using POD to reduce the flux density data, the number of rows was reduced to only 12 in the heavily reduced case. With tighter tolerance for the POD forming, the number of rows is higher, but still a lot less than in the full data case.

The number of homogeneous regions identified in the full data case is the same as the number of integration points, because no combined regions were identified. In the heavily reduced case, 39 homogeneous regions were identified. When the tolerances were tightened, the number of regions increased.

By adjusting the tolerances of both the POD forming and the homogeneous region identifying, the performance of the simulation can be adjusted. As seen in the results shown here, it is possible to get more accurate results with tighter tolerances, but it will make the simulation to take more time to complete. The optimal tolerances can then be determined depending on which is more important, the result accuracy or the simulation speed.

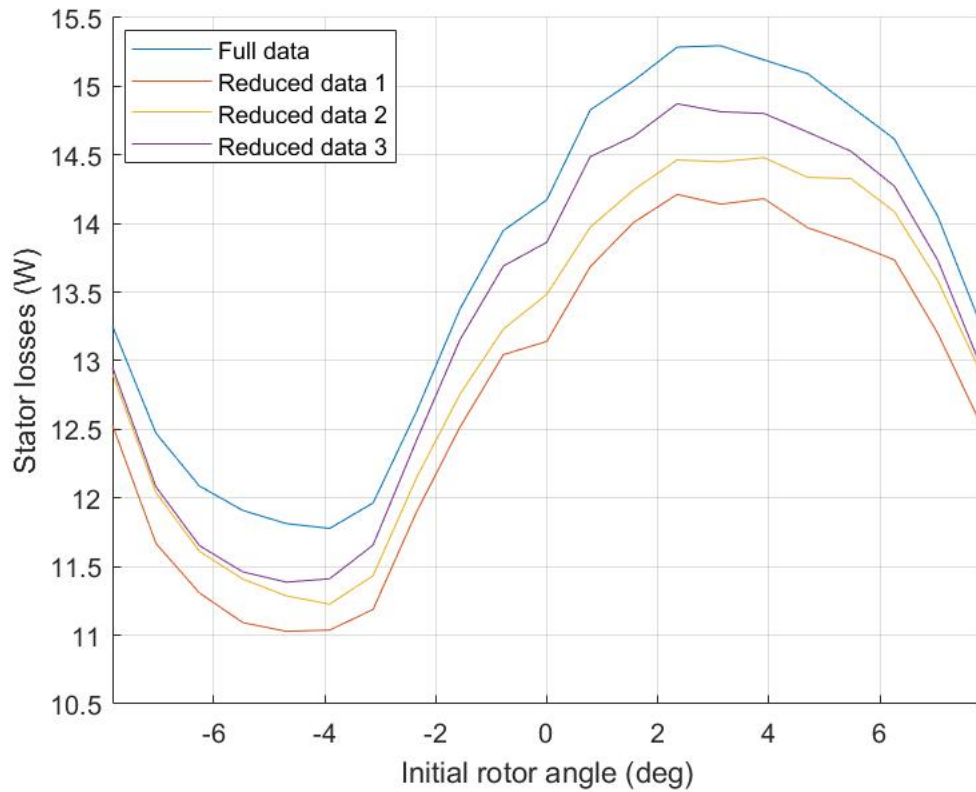


Figure 4.8. Stator losses as a function of the initial rotor angle in four sets of simulations with different tolerance settings.

	Full data	Reduced data 1	Reduced data 2	Reduced data 3
Average Simulation time (s)	52,7	1,5	4,6	4,9
Total Simulation time (min)	18,4	0,5	1,6	1,7
Number of rows in flux-density distribution	3331	12	338	338
Number of homogeneous regions	3331	39	80	218

Table 4.4. Simulation times and data sizes.

5. CONCLUSION

The objective of the thesis was to create a faster way for time domain simulation of TFMS than time domain 3D FEM simulations. For this reason, a MATLAB/Simulink model of a TFMS based on LUTs was developed. The simulink model was based on utilizing precalculated data stored on LUTs.

To generate the precalculated data, a simple 3D FEM model of a TFRM was modelled with COMSOL Multiphysics software. The data to the LUTs was produced by simulating a series of static 3D FEM simulations using the COMSOL model. The necessary data to be stored to the LUTs, were the stator winding flux linkage, electromagnetic torque and magnetic flux density distribution as a function of current and rotor angle. With this data, it was then possible to simulate time domain simulations with the Simulink model.

The basic idea was that by simulating a quite lengthy series of FEM simulations to create the data, it would then be possible to do to many time domain simulations with Simulink very quickly by utilizing the precalculated data. Whereas, if the time domain simulations were to be solved using the FEM model, each simulation would take a very long time.

The objective of creating LUT based Simulink model for fast time domain simulations was reached. The simulations with Simulink took only seconds to finish, while the simulations with COMSOL could take tens of minutes to finish. The time domain Simulink simulations also seemed to produce accurate results when compared with time domain COMSOL simulations.

Including the iron core loss calculations to the Simulink model made the simulations slower when compared to simulations without loss calculations. It was possible to make these simulations faster by using the data reduction methods. Proper orthogonal decomposition was used to reduce the amount of stored data. Also, homogeneous regions were identified from the magnetic flux density distribution. These methods made the simulations significantly faster with a slight decrease in accuracy.

REFERENCES

- [1] Nasiri-Zarandi, R., Ghaheri, A. and Abbaszadeh, K. Thermal Modeling and Analysis of a Novel Transverse Flux HAPM Generator for Small-Scale Wind Turbine Application. *IEEE transactions on energy conversion* 35.1 (2020), pp. 445–453.
- [2] Jia, Z., Lin, H., Yang, H. and Mi, C. Transverse Flux Permanent Magnet Motor with Double-C Stator Hoops and Flux-Concentrated Rotor for In-Wheel Drive Electric Vehicle. *IEEE Energy Conversion Congress and Exposition (ECCE)* (2014), pp. 4804–4808.
- [3] Takizawa, D., Takahashi, T., Shimizu, H. and Kato, R. Development of Transverse Flux Motor with Improved Material and Manufacturing Method. *SAE International Journal of Passenger Cars - Electronic and Electrical Systems* 6 (2013), pp. 366–376.
- [4] Pompermaier, C., Sjöberg, L. and Nord, G. Design and optimization of a Permanent Magnet Transverse Flux Machine. *2012 XXth International Conference on Electrical Machines* (2012), pp. 606–611.
- [5] Rasilo, P., Lemesle, M.-A., Belahcen, A., Arkkio, A. and Hinkkanen, M. Comparison of Finite-Element Based State-Space Models for PM Synchronous Machines. *IEEE transactions on energy conversion* 29.2 (2014), pp. 535–543.
- [6] Pinto, D. E., Pop, A.-C., Kempkes, J. and Gyselinck, J. *Reduced-Order Eddy-Current Loss Modelling of Electrical Machines Using Lookup Tables Obtained via 2D Finite-Element*. 2018.
- [7] Boldea, I. *Linear Electric Machines, Drives, and MAGLEVs Handbook*. Taylor & Francis Group, 2013. 652 p.
- [8] Le Doeuff, R. and El Hadi Zaïm, M. *Rotating Electrical Machines : From Matrix Modeling to Implementation*. John Wiley & Sons, Incorporated, 2010. 305 p.
- [9] Hughes, A. and Drury, B. *Electric motors and drives : fundamentals, types and applications*. Elsevier, 2013. 458 p.
- [10] Nerg, J., Rilla, M., Ruuskanen, V., Pyrhonen, J. and Ruotsalainen, S. Direct-Driven Interior Magnet Permanent-Magnet Synchronous Motors for a Full Electric Sports Car. *IEEE transactions on industrial electronics* 61.8 (2014), pp. 4286–4294.
- [11] Villet, W. T. and Kamper, M. J. Variable-Gear EV Reluctance Synchronous Motor Drives—An Evaluation of Rotor Structures for Position-Sensorless Control. *IEEE transactions on industrial electronics* 61.10 (2014), pp. 5732–5740.
- [12] Rabiei, A., Thiringer, T., Alatalo, M. and Grunditz, E. A. Improved Maximum-Torque-Per-Ampere Algorithm Accounting for Core Saturation, Cross-Coupling Effect, and

- Temperature for a PMSM Intended for Vehicular Applications. *IEEE transactions on transportation electrification* 2.2 (2016), pp. 150–159.
- [13] Haque, M., Negnevitsky, M. and Muttaqi, K. A Novel Control Strategy for a Variable-Speed Wind Turbine With a Permanent-Magnet Synchronous Generator. *IEEE transactions on industry applications* 46.1 (2010), pp. 331–339.
- [14] Dehghan, S., Mohamadian, M. and Varjani, A. A New Variable-Speed Wind Energy Conversion System Using Permanent-Magnet Synchronous Generator and Z-Source Inverter. *IEEE transactions on energy conversion* 24.3 (2009), pp. 714–724.
- [15] Ghosh, S. *Electrical machines*. Pearson India, 2012. 848 p.
- [16] Baggini, A. and Sumper, A. *Electrical energy efficiency technologies and applications*. John Wiley & Sons, Incorporated, 2012. 420 p.
- [17] Gieras, J. F. *Permanent magnet motor technology : design and applications*. CRC Press LLC, 2010. 618 p.
- [18] Luk, P. C.-K., Abdulrahem, H. A. and Xia, B. Low-cost high-performance ferrite permanent magnet machines in EV applications: A comprehensive review. *ETRANSPORTATION* 6 (2020).
- [19] Bianchi, N., Bolognani, S., Carraro, E., Castiello, M. and Fornasiero, E. Electric Vehicle Traction Based on Synchronous Reluctance Motors. *IEEE transactions on industry applications* 52.6 (2016), pp. 4762–4769.
- [20] Pellegrino, G., Jahns, T. M., Bianchi, N., Soong, W. L. and Cupertino, F. *The Rediscovery of Synchronous Reluctance and Ferrite Permanent Magnet Motors*. Springer Cham, 2016. 136 p.
- [21] Babetto, C., Bacco, G. and Bianchi, N. Synchronous Reluctance Machine Optimization for High-Speed Applications Reluctance Motors. *IEEE transactions on energy conversion* 33.3 (2018), pp. 1266–1273.
- [22] Jones, G. R. *Electrical Engineer's Reference Book, 15th Edition*. Newnes, 2013. 1376 p.
- [23] Kaiser, B. and Parspour, N. Transverse Flux Machine—A Review. *IEEE access* 10 (2022), pp. 18395–18419.
- [24] Asgari, S. and Mirsalim, M. A Novel Dual-Stator Radial-Flux Machine With Diametrically Magnetized Cylindrical Permanent Magnets. *IEEE transactions on industrial electronics* 66.5 (2019), pp. 3605–3614.
- [25] De Donato, G., Capponi, F. G., Borocci, G., Caricchi, F., Beneduce Luigi ; Fratelli, L. and Tarantino, A. Ω -Shaped Axial-Flux Permanent-Magnet Machine for Direct-Drive Applications With Constrained Shaft Height. *IEEE transactions on industry applications* 51.4 (2015), pp. 3050–3058.
- [26] Bang, D.-J., Polinder, H., Shrestha, G. and Ferreira, J. Design of a Lightweight Transverse Flux Permanent Magnet Machine for Direct-drive Wind Turbines. *2008 IEEE Industry Applications Society Annual Meeting* (2008), pp. 1–7.

- [27] Guo, Y., Zhu, J. G., Watterson, P. and Wu, W. Development of a PM Transverse Flux Motor With Soft Magnetic Composite Core. *IEEE transactions on energy conversion* 21.2 (2006), pp. 426–434.
- [28] Kruse, R., Pfaff, G. and Pfeiffer, C. Transverse Flux Reluctance Motor for Direct Servodrive Applications. *Conference Record of 1998 IEEE Industry Applications Conference 1* (1998), pp. 655–662.
- [29] Arkkio, A. *Analysis of Induction Motors Based on the Numerical Solution of the Magnetic Field and Circuit Equations*. 1987.
- [30] *COMSOL Documentation*. URL: <https://doc.comsol.com/6.0/docserver/#!/com.comsol.help.comsol/helpdesk/helpdesk.html>.
- [31] *MATLAB Documentation*. URL: <https://se.mathworks.com/help/> (visited on 04/25/2023).
- [32] Henneron, T. and Clenet, S. Model Order Reduction of Non-Linear Magnetostatic Problems Based on POD and DEI Methods. *IEEE transactions on magnetics* 50.2 (2014), pp. 33–36.

APPENDIX A: SIMULINK MODEL

