Finite Element Analysis of a 6-phase BLDC Reluctance Machine

by
Milos Lukic

Submitted to the Department of Electrical Engineering, Electronics,
Computers and Systems
in partial fulfillment of the requirements for the degree of
Erasmus Mundus Master Course in Sustainable Transportation and
Electrical Power Systems
at the
UNIVERSIDAD DE OVIEDO
September 2017

Author ……………………………………………………………………………………

Certified by ………………………………………………………………………………...

Giulio De Donato
Assistant Professor (Sapienza – University of Rome)
Thesis Supervisor

Certified by…………………………………………………………………………………

Ion Boldea
Full Professor (Polytechnic University of Timisoara)
Thesis Co-advisor

Finite Element Analysis of a 6-phase BLDC Reluctance Machine

by

Milos Lukic

Submitted to the Department of Electrical Engineering, Electronics, Computers and Systems
on September 1, 2017, in partial fulfillment of the
requirements for the degree of
Erasmus Mundus Master Course in Sustainable Transportation and Electrical Power Systems

Abstract:

Multi-phase BLDC reluctance machines are a class of synchronous machines that was initially proposed in the eighties, although it didn’t receive much attention due to the concurrent success of high energy-product, rare-earth permanent magnet synchronous machines. Renewed attention has been drawn towards this class of machines very recently, due to the spike in the cost of rare-earth permanent magnets that occurred in 2011. Until now, only a few research groups have investigated BLDC reluctance machines, hence this thesis is intended as a contribution to a field of research which is covered by a limited amount of papers in the scientific literature. After an in-depth introduction, the thesis covers the preparation of the FEA model in MagNet 7. Matlab scripts have been developed to act as interfaces with MagNet to parametrise the model and to allow an easy exchange of the machine’s parameters and FEA results. Simulations are then reported, reporting torque and flux in various operating conditions, followed by the calculation of significant machine parameters such as inductances (apparent and incremental) and resistances. Moreover, the behaviour of the machine is investigated by applying various field and torque currents and by observing changes in the machine performances. Torque comparison between the model with non-linear and linear materials is also investigated for various field and torque currents. The effect of rotor skewing is demonstrated as a means of reducing torque ripple.

Thesis supervisor: Giulio De Donato
Title: Assistant professor (Sapienza – University of Rome)

Thesis Co-advisor: Ion Boldea
Title: Full professor (Polytechnic University of Timisoara)

Acknowledgment

First of all, I would like to thank for the opportunity to develop my master thesis at Sapienza – University of Rome, officially the best and biggest Italian University. I am truly grateful working with thesis supervisor, Giulio De Donato. His daily availability and prompt replies to my quandaries are highly appreciated. Moreover, his advices and suggestions were priceless in the thesis development, from the beginning of the project even to the writing stage. Also, thank to Ion Boldea for his guidelines. I would also like to thank to all professors from STEPS consortium for their unselfish share of knowledge and help during this 2 years.

Furthermore, I would like to thank all my classmates from Sustainable Transportation Track for incredible collegiality. I will never forget fruitful discussions we had and your altruism. A special thank goes to Diego and Froy, now friends for life, not only for their support during studies but also for making these two years an unforgettable experience.

In the end, I would like to thank my family for their unconditional support and love during this master course.

Table of Contents

Introduction 15

1 State of The Art Review

1.1 Operation of Separately Excited DC Machines ………………………………. 19

1.2 Principle of Operation of multi-phase BLDC Reluctance Machines ……….... 21

1.3 Control Aspects of multi-phase BLDC Reluctance Machine ………………… 26

1.4 Advantages and Disadvantages ……………………………………………….. 29

2 Finite Element Modelling of the Machine

2.1 Scripting and Connection of Matlab and MagNet ……………………………. 33

2.2 Parametric Drawing of the Machine via Matlab Scripts ……………………… 34

2.3 Component Creation and Assignment of the Materials ………………………. 46

2.4 Coils, Currents and Motion Components ……………………………………... 53

2.5 Mesh Generation and Boundaries …………………………………………….. 60

3 Simulations and Results

3.1 Analysis of the Machine with Various Field (I_F) and Torque (I_T) Currents.. 65

3.1.1 Field current 12A, torque current 12A …………………………………. 65

3.1.2 Field current 6A, torque current 6A ……………………………………. 69

3.1.3 Armature Reaction ……………………………………………………... 71

3.1.3.1 Field current 12A, torque current 0A ………………………….. 71

3.1.3.2 Field current 6A, torque current 0A …………………………… 75

3.1.4 Average Torque and Ripple ……………………………………………. 77

3.2 Parameter Identification ……………………………………………………… 81

3.2.1 Resistance ……………………………………………………………… 81

3.2.2 Inductances …………………………………………………………….. 82

3.2.2.1 Calculation of Apparent Inductances ………………………….. 85

3.2.2.2 Calculation of Incremental Inductances ($\frac{\partial\psi}{\partial i}$) ………………….. 88

3.2.2.3 Calculation of Self-emf ($\frac{\partial\psi}{\partial\theta_r}$) ………………………………….. 90

3.2.2.4 Calculation of $\frac{d\psi}{dt}$ ……………………………………………… 92

3.2.3 Torque Calculation Using Apparent Inductances ……………………… 94

3.3 Effect of The Air Gap Length ………………………………………………… 98

3.4 Behaviour of The Machine versus Speed …………………………………….. 102

4 Summary and Future Development

4.1 Summary and Conclusions …………………………………………………… 106

4.2 Future Developments ………………………………………………………… 107

4.3 Quality Report ………………………………………………………………… 107

5 Bibliography 109

List of Figures

Name	Description	Page
Fig. I.1	A classification of motors	15
Fig. I.2	Prices of Neodymium and Dysprosium metal during 5-year period from Jan. 2009 and July 2011, [3]	16
Fig. 1.1	Cross-section of the stator with (a) DC current excitation coil and (b) PM excitation	19
Fig. 1.2	Rotor currents distribution	20
Fig. 1.3	Explanation of the commutation process	20
Fig. 1.4	Appearance of the collector	20
Fig. 1.5	(a) Stator field (b) Rotor field (c) Resultant field	21
Fig. 1.6	Exciter-less DC machine with brushes moved from neutral axis	22
Fig. 1.7	One configuration of six phase BLDC reluctance machine	22
Fig. 1.8	Square wave current of six-phase BLDC reluctance machine	23
Fig. 1.9	Trapezoidal currents of six-phase BLDC reluctance machine proposed by (a) Lipo and (b) Boldea	23
Fig. 1.10	Rotor with three flux barriers	24
Fig. 1.11	Two different types of lamination: (a) radial (b) axial	24
Fig. 1.12	Assembled 7.5kW ALA rotor before turning and grinding operations	24
Fig. 1.13	Another configuration of 6 phase BLDC reluctance machine	25
Fig. 1.14	Definition of q and d axis	25
Fig. 1.15	Field flux in the direction of d-axis	26
Fig. 1.16	Full-bridge converter	26
Fig. 1.17	Half-bridge converter	27
Fig. 1.18	Control of each phase with half-bridge converter and presence of null conductor	27
Fig. 1.19	Drive configuration proposed by Lipo and Law	27
Fig. 1.20	Block diagram of current control proposed by Kamper	28
Fig. 1.21	Three industrial inverters scheme for control of six-phase machine	28
Fig. 2.1	Flowchart of the model preparation in MagNet	31
Fig. 2.2	Six-phase BLDC reluctance machine	32
Fig. 2.3	Construction slice	35
Fig. 2.4	Stator geometry and its parameters	35
Fig. 2.5	Point enumeration for stator drawing	36
Fig. 2.6	Half of the tooth – first stage in stator drawing	39
Fig. 2.7	Stage after for loop	40
Fig. 2.8	Completed stator of the machine	40
Fig. 2.9	Final stator version. Rotated structure from Fig. 3.6. for 30 degrees	40
Fig. 2.10	Stator with same parameters as in table 2, but with $N_s = 6$.	41
Fig. 2.11	Stator with $N_s = 6, t_w = 15\text{mm}$ and $s_o = 50\text{mm}$	41
Fig. 2.12	Rotor geometry and its parameter	42
Fig. 2.13	Stage in rotor drawing	42
Fig. 2.14	Point enumeration for drawing of rotor’s ribs	42
Fig. 2.15	Final structure	45
Fig. 2.16	Possible combinations of rotor lamination	45
Fig. 2.17	Construction slice surfaces of the model	46
Fig. 2.18	Construction slice surfaces in the air gap	47

Fig. 2.19	Model when sweeping is applied	47
Fig. 2.20	Shape of matrix M from equation (2.2)	48
Fig. 2.21	Three different normal vectors which define rotor lamination	49
Fig. 2.22	Uniform direction type	49
Fig. 2.23	Small piece of lamination	49
Fig. 2.24	$B-H$ curves of bulk M300-35A, parallel and normal direction	50
Fig. 2.25	Coil numeration	54
Fig. 2.26	Current waveform in rated conditions $I_F = I_T = 12A$	54
Fig. 2.27	Current waveform with $I_F = 12A$, $I_T = 6A$	55
Fig. 2.28	Current waveform with $I_F = 6A$, $I_T = 12A$	55
Fig. 2.29	Auxiliary current from which all other currents are obtained	56
Fig. 2.30	Auxiliary current and current of phase 5	56
Fig. 2.31	Time instants necessary for definition of auxiliary current	56
Fig. 2.32	Mesh elements in one of the simulations	61
Fig. 2.33	Odd periodicity	61
Fig. 3.1	Currents with $(I_F, I_T) = (12{,}12)A$	66
Fig. 3.2	Flux linkages of phase 1 to phase 6	66
Fig. 3.3	Torque waveform for $(I_F, I_T) = (12{,}12)A$	67
Fig. 3.4	Voltage waveform of phase 1	67
Fig. 3.5	Instantaneous electrical power	67
Fig. 3.6	Currents with $(I_F, I_T) = (12{,}6)A$	69
Fig. 3.7	Flux linkages of phase 1 to phase 6	69
Fig. 3.8	Torque waveform for $(I_F, I_T) = (12{,}6)A$	70
Fig. 3.9	Instantaneous electrical power	70
Fig. 3.10	Currents with $(I_F, I_T) = (12{,}0)A$	72
Fig. 3.11	Currents of the machine with irregular commutation	72
Fig. 3.12	Flux linkages in no-load condition with $I_F = 12A$	72
Fig. 3.13	Torque waveform with $(I_F, I_T) = (12{,}0)A$	73
Fig. 3.14	Position of the rotor at 20ms	73
Fig. 3.15	Flux lines at 20 ms with no-load conditions	74
Fig. 3.16	Flux lines under load $I_T = 12A$	74
Fig. 3.17	No-load flux density in the middle of the air gap	74
Fig. 3.18	No-load and full-load ($I_T = 12A$) flux densities in the middle of the air gap	75
Fig. 3.19	Currents with $(I_F, I_T) = (6{,}0)A$	75
Fig. 3.20	No-load flux densities with $I_F = 6A$	76
Fig. 3.21	Torque waveform with $(I_F, I_T) = (6{,}0)A$	76
Fig. 3.22	No-load flux density wit $I_F = 6A$	76
Fig. 3.23	Flux densities with $I_F = 6A$, and load current of 6A and 0A.	77
Fig. 3.24	Torque as a function of I_T with $I_F = 12A$	78
Fig. 3.25	Torque as a function of I_T with $I_F = 10A$	78
Fig. 3.26	Torque as a function of I_T with $I_F = 8A$	79
Fig. 3.27	Torque as a function of I_T with $I_F = 6A$	79
Fig. 3.28	Torque as a function of I_T with$I_F = 4A$	79
Fig. 3.29	Torque as a function of I_T with$I_F = 2A$	80
Fig. 3.30	Torque as a function of I_T for various field currents	80
Fig. 3.31	Torque waveform with $(I_F, I_T) = (8{,}4)A$	81

Fig. 3.32	Torque waveform with $(I_F, I_T) = (8{,}4)A$ without and with rotor skewing	81
Fig. 3.33	Dimensions of the slot	82
Fig. 3.34	$\psi - i$ curve, apparent and incremental inductances	84
Fig. 3.35	Equivalent circuit of the equation (3.7) for phase 1	86
Fig. 3.36	Apparent inductances of phase 1	86
Fig. 3.37	Apparent inductances L_{11} and L_{22}	87
Fig. 3.38	Apparent inductances L_{31} and L_{42}	87
Fig. 3.39	Apparent inductances L_{21} and L_{21}	87
Fig. 3.40	Apparent self-inductances	88
Fig. 3.41	Current of phase 1, rated and with disturbances	89
Fig. 3.42	Currents i_{1_plus}, $i_{1_{minus}}$ and rated currents of phases 2 to 6	89
Fig. 3.43	Incremental inductances L_{i1}	89
Fig. 3.44	Apparent and incremental self-inductance of phase 1	90
Fig. 3.45	Problem of having different values of currents in two consecutive time steps	91
Fig. 3.46	Solution of the problem with shifting current	91
Fig. 3.47	Partial or self-emf of phase 1	91
Fig. 3.48	Self-emfs of phase 1 and 2	92
Fig. 3.49	Voltage of coil 1 calculated as $\frac{d\psi}{dt}$	92
Fig. 3.50	Voltage from MagNet and calculated as $\frac{d\psi}{dt}$	93
Fig. 3.51	MagNet and voltage calculated using parameters through equation (3.6)	93
Fig. 3.52	Incremental L_{11} inductance in load and no-load conditions	94
Fig. 3.53	Comparison of torque calculated with FEM and with linear formula with $I_F = 12A$	96
Fig. 3.54	Comparison of torque calculated with FEM and with linear formula with $I_F = 6A$	97
Fig. 3.55	Comparison of torque calculated with FEM and with linear formula with $I_F = 2A$	97
Fig. 3.56	Comparison of torque calculated with FEM and with linear formula for field currents of 12A,6A and 2A	97
Fig. 3.57	Machine with 0.35 *mm* air gap	98
Fig. 3.58	Machine with 0.6 *mm* air gap	98
Fig. 3.59	Machine with 1 *mm* air gap	99
Fig. 3.60	Machine with 1.5 *mm* air gap	99
Fig. 3.61	No load flux density with $I_F = 12A$ and 4 different air gaps values	100
Fig. 3.62	No load flux density with $I_F = 6A$ and 4 different air gaps values	100
Fig. 3.63	Rotor position at 20ms	100
Fig. 3.64	No-load and full-load ($I_T = 12A$) flux densities with the air gap of 0.35 *mm*	101
Fig. 3.65	No-load and full-load ($I_T = 12A$) flux densities with the air gap of 0.6 *mm*	101
Fig. 3.66	Power losses $(P_{hyst\_stator} + P_{eddy\_stator} + P_{Cu\_insul} + P_{hyst_{rotor}} + P_{hyst\_rotor})$ as function of speed for $(I_F, I_T) = (12{,}12)A$	104
Fig. 3.67	Power losses $(P_{hyst\_stator} + P_{eddy\_stator} + P_{Cu\_insul} + P_{hyst_{rotor}} + P_{hyst\_rotor})$ as function of speed for $(I_F, I_T) = (12{,}6)A$	104

List of Tables

Name	Description	Page
Table 2.1	Main parameters of the machine	32
Table 2.2	Geometric parameters of the stator	32
Table 2.3	Some of the rotor's geometric parameters	32
Table 2.4	Missing parameters (degrees of freedom) of the rotor's geometry	41
Table 2.5	Materials assigned to various parts of the machine	47
Table 2.6	Maximum mesh element size for each part of the machine	60
Table 3.1	Some machine's results $(I_F, I_T) = (12{,}12)A$	67,68
Table 3.2	All losses in the machine with $(I_F, I_T) = (12{,}12)A$ and 250rpm	68,69
Table 3.3	Some machine's results $(I_F, I_T) = (12{,}6)A$	70
Table 3.4	All losses in the machine with $(I_F, I_T) = (12{,}6)A$ and 250rpm	71
Table 3.5	Average torque values for $I_F = 12A$ for 4 different values of the air gap	101
Table 3.6	Average torque values for $I_F = 6A$ for 4 different values of the air gap	101
Table 3.7	Efficiency of the machine for $(I_F, I_T) = (12{,}12)A$ and different speeds	102
Table 3.8	Efficiency of the machine for $(I_F, I_T) = (12{,}6)A$ and different speeds	102
Table 3.9	Stator hysteresis, eddy current losses and ohmic losses in insulation for $(I_F, I_T) = (12{,}12)A$ for various speeds	102,103
Table 3.10	Rotor hysteresis and eddy current losses for $(I_F, I_T) = (12{,}12)A$ for various speeds	103
Table 3.11	Stator hysteresis, eddy current losses and ohmic losses in insulation for $(I_F, I_T) = (12{,}6)A$ for various speeds	103
Table 3.12	Rotor hysteresis and eddy current losses for $(I_F, I_T) = (12{,}6)A$ for various speeds	103

Introduction

I.1 Project outline

The importance of electromechanical energy conversion is undisputable in today's world. Electrical machines are essential part in electrical energy production, without which world we know today wouldn't exist. Moreover, around 60% of the produced electrical energy is consumed by electrical machines, while the rest by other electrical loads (heating, lightning etc.) [1]. This isn't surprising since in industry roll-mills, conveyers, elevators, ventilators, pumps etc., are required for various industrial process, like in industries of metal processing. Moreover, modern factories nowadays have automated their production with industrial robots. In both cases, electrical machines are required for a controlled motion. Contrary, to industrial applications electrical machines can also be found in devices we use every day such as washing machines, hair dryers, PCs etc. In transportation, electrical machines are used in trams and trains for decades, but nowadays due to environmental reasons electrical machines are finding use in electric and hybrid vehicles as well as in airplanes and ships. Over almost two centuries, many different types of electrical machines have been developed. The most important types can be seen in the schematic in Fig. I.1 [2].

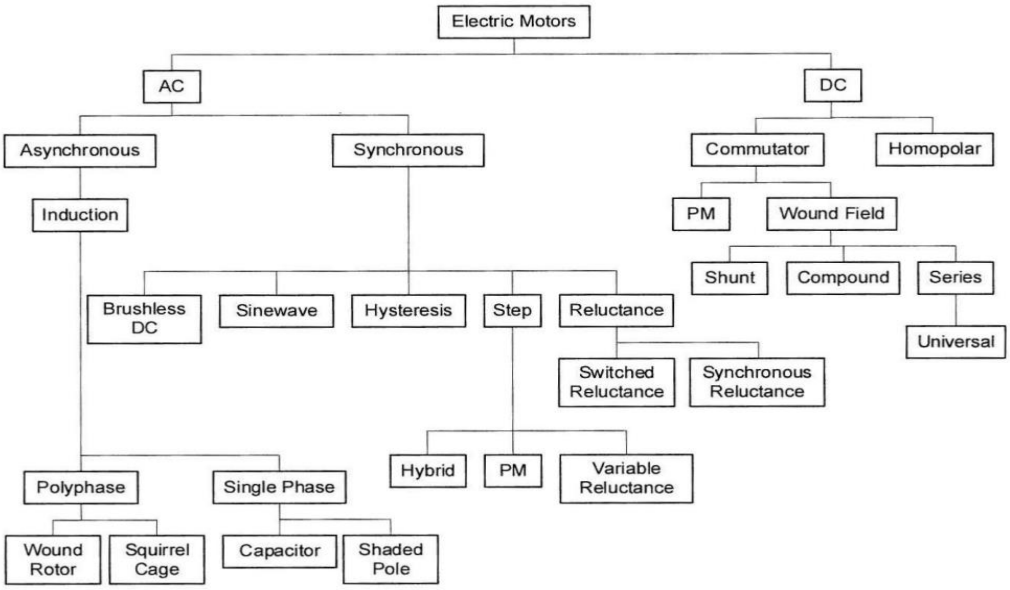


Fig. I.1 A classification of motors

With the discovery of new materials and advancements in power electronics, opportunities in machine design have increased drastically. Fifty years ago, there were mainly three types available commercially: induction machines (IM), synchronous machines (SM) (mainly as generators) and DC machines. If we compare the market and available models 50 years ago and nowadays significant increase can be observed. Some of the new models are: Synchronous PM machines, Synchronous Reluctance machines, Switched Reluctance machine etc. Maybe one of the most significant moments was in 1980s, the production of high energy neodymium-iron-boron material (NdFeB). That allowed mass production of synchronous PM machines which were dominant many years due to high efficiency, power density etc. However, in 2011, when the price of Neodymium and Dysprosium (metal which allows higher Currie temperature) increased, production of these high-performance machines was effected. In the Fig. I.2 we can see how price rose 25 times for Neodymium and 22 times Dysprosium in the period from Jan 09 and Jul 11. [3]

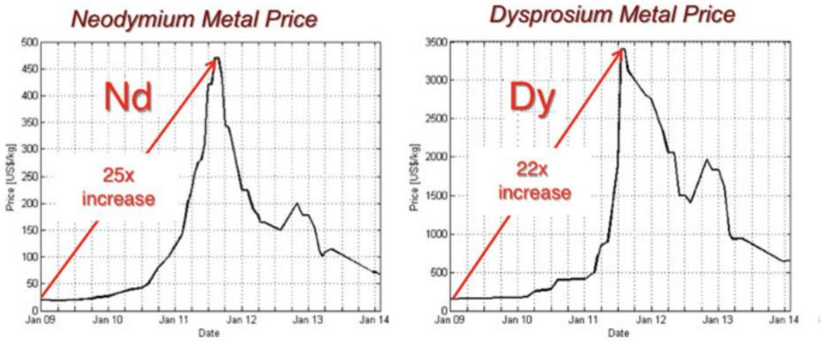


Fig. I.2 Prices of Neodymium and Dysprosium metal during 5-year period from Jan. 2009 and July 2011, [3]

Despite the decrease after the peak, manufacturers and universities started to explore new alternative types of machines which either don't use permanent magnets at all or use ferrite magnets (less expensive option than NdFeB). Among these, one of the interesting solutions is the multi-phase brushless DC (BLDC) reluctance machine. This machine is not ne: it was first proposed by Weh and Schroder in 1984, but since in that period NdFeB was gaining popularity, not much attention was paid to this kind of machine. However, in 1990 Boldea revived this concept followed by Lipo and Law in 1992. In recent years, Kamper contributed to this field also.

I.2 Chapter overview

In Chapter 1, the state of the art on multi-phase BLDC reluctance machine is presented. The principle of operation of this type of the machine is explained. Moreover, the control strategies for driving multi-phase BLDC reluctance machine are discussed and converter topologies are presented. Also, some pros and cons of this machine are presented.

Chapter 2 describes in detail the Finite Element (FE) model preparation in MagNet and Matlab. Before the modelling procedure explanation, parameters of the machine are presented and scripting, basic commands and connection between two programs are explained. The process of drawing machine's edges, selection and modelling of the used materials, definition of coils, currents, mesh and boundary conditions are explained in depth. Each part of the modelling process is followed by Matlab script and its explanation.

Various FE analyses have been performed in MagNet and their description and results are presented in Chapter 3. Equivalent per-phase circuit parameters are presented and limited use of these parameters is explained. Also, average and ripple torque were calculated for 36 pairs of field and torque current. Armature reaction is investigated i.e. how load currents affects flux density in the air gap. One set of simulations was dedicated to airgap length change and how this affects torque and flux in the machine. In the end power losses are calculated for various speeds and various field current levels.

In the final chapter, Chapter 4, thesis is briefly summarized and future research possibilities are presented.

Chapter 1

State of the Art Review

As mentioned before, the concept of the multi-phase BLDC reluctance machine was first introduced by Weh [4] and he explained the functioning of the machine as that of an electronically commutated DC machine with both field and armature windings placed in the stator. Hence, operation of the BLDC reluctance machine is similar to that of a separately excited DC machine and thus it is convenient to briefly describe the operation of the DC machine first. Then, the principle of the operation of multi-phase BLDC reluctance machine is explained in detail, followed by proposed control strategies and description of power converters. In the end, advantages and disadvantages and comparison with other types of the machines are presented.

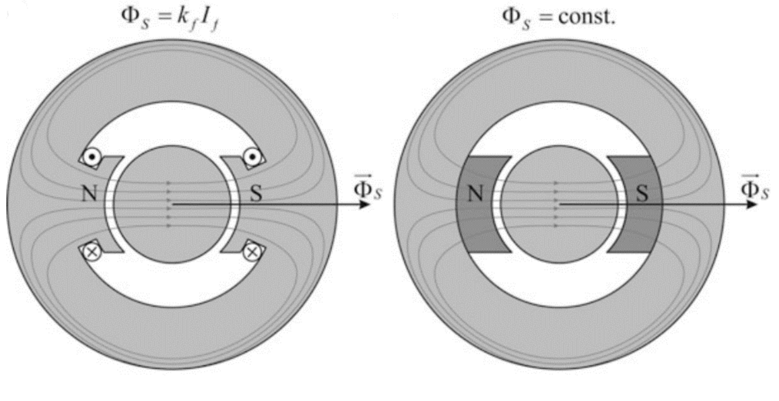
1.1. Operation of Separately Excited DC Machine

It is well known that torque production in electrical machines is related to the cross product of stator and rotor flux

$$T_{em} = k|\psi_r \times \psi_s| = k\psi_r\psi_s \sin\big(\measuredangle(\psi_r, \psi_s)\big) \tag{1.1}$$

A peculiarity of DC machines is that by construction, angle between those fluxes is 90° (maximizing the value of cross product) and that they are not rotary, i.e. they are static. These two features are obtained with the mechanical commutator, which will be explained in detail later.

A cross section of the stator of the DC machine is shown in Fig. 1.1. In Fig. 1.1(a) the stator winding is present and it produces stator field (flux) which is dependent on the current through the winding. On the other hand, in Fig. 1.1(b) the stator field (flux) is produced by permanent magnets and it is constant. Since the focus of this thesis is on PM-less machine, only the machine with DC current stator winding will be explained. The stator winding is also called the *excitation winding* or *field winding* and it produces the *excitation field*.



(a) (b)

Fig. 1.1 Cross-section of the stator with (a) DC current excitation coil and (b) PM excitation

What can be noticed from the Fig. 1.1 is that the stator flux is static and it always has the same direction. In order to produce a non-zero average torque, the rotor field also must be static in respect to the stator field. Conductors which carry rotor current are embedded In the rotor of DC machines conductors are embedded in rotor of DC machines. In order to create rotor field which is 90° respect to stator field, rotor currents must always be in the direction shown in Fig. 1.2, independently of the rotor position.

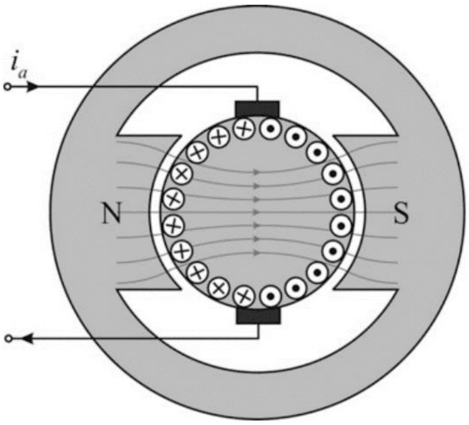
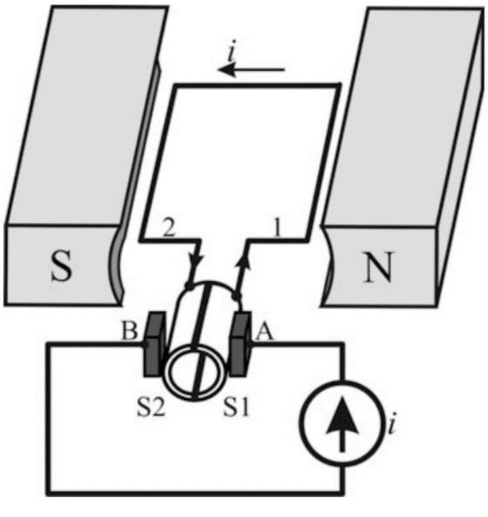


Fig. 1.2 Rotor currents distribution

The mechanical commutator is a device which is enabling rotor flux to be static and its simple version is shown in Fig. 1.3 while a real one is shown in Fig. 1.4. Rotating part of the commutator is called *collector* and it is divided into *segments* (in this case it has only two segments S1 and S2). Segments are separated between each other by insulation among themselves with insulation. The static part of the commutator has carbon *brushes* (A and B). The rotor DC current, called *armature current*, is fed from an external source via the brushes and collector to the rotor conductors.



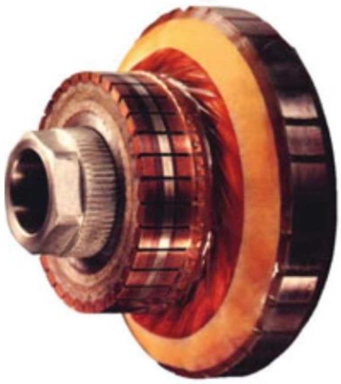


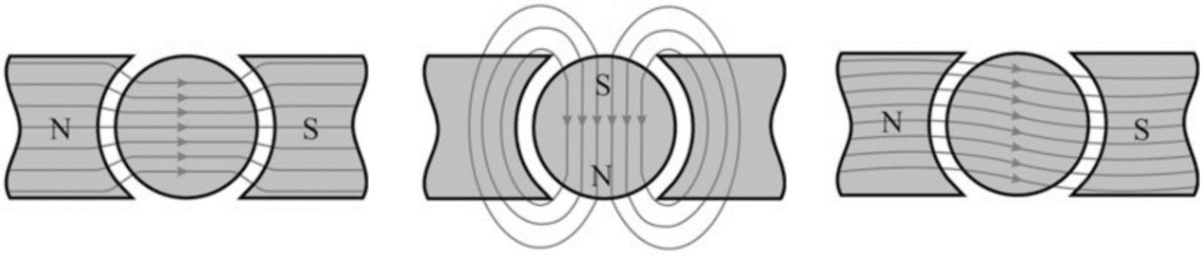
Fig. 1.3 Explanation of the commutation process

Fig 1.4 Appearance of the collector

From Fig. 1.3 it can be seen that conductor 1 after half a revolution, will be at the previous position of conductor 2 with current leaving brushes. Hence, the current in conductor 1 has changed direction, but the current distribution seen from the stator has remained the same. Although it has been a viable solution for many years the mechanical commutator has two main disadvantages:

- Wearing of the carbon
- Sparks which happen during commutation due to short circuiting of rotor conductors

Another important aspect which must be considered in DC machines is effect of armature current (MMF). As seen previously, stator currents create stator field in horizontal direction, Fig. 1.5(a), while rotor currents create field in vertical direction, Fig. 1.5(b). Total field in the machine is sum of these fields and it is presented in Fig. 1.5 (c). Even though rotor flux is small, due to high reluctance in its direction, it is sufficient to cause distortion of stator field as it can be seen in Fig. 1.5(c). This effect is called armature reaction. Armature reaction affects air gap flux distribution as well as magnitude. Flux distribution is important for commutation process, while flux magnitude directly influences produced torque [5].



(a) (b) (c)

Fig. 1.5 (a) Stator field (b) Rotor field (c) Resultant field

1.2. Principle of Operation of multi-phase BLDC Reluctance Machines

After this brief reminder, we can continue with the explanation of the principles of operation of the machine which is focus of this thesis. According to Boldea [6], as shown in Fig. 1.6, the operation of a six-phase BLDC reluctance machine can be explained if we consider an exciter-less DC brush machine with a six coil rotor, with brushes rotated from the neutral axis to the corner of the stator poles, and with an enlarged magnetic saliency in the stator, produced by insertion of flux barriers. Coils E-E’ and F-F’ are producing field in *d*-direction, hence they are called *field windings* whilst coils A-A’, B-B’, C-C’ and D-D’ produce field in *q*-direction and hence they are called *torque (armature) windings*. This machine still has the properties of the DC machine (fluxes are 90 degrees apart and they are static because of the brushes), but, as Law stated in [7], the torque and field currents are not independent anymore and commutation with this position of brushes will cause even more sparks than in the normal DC machine configuration and thus significant damage. Therefore, the coils are moved from the rotor to the stator.

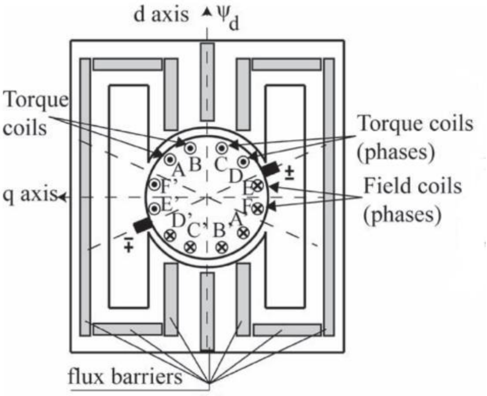


Fig. 1.6 Exciter-less DC machine with brushes moved from neutral axis

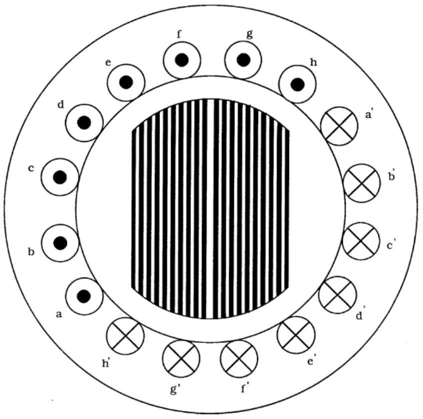


Fig 1.7 One configuration of six phase BLDC reluctance machine

If the coils are moved to the stator and the rotor is made salient, as in Fig. 1.7, the machine will operate in the same manner. However, since the coils are now in the stator, there is no longer any need for a mechanical commutator anymore; consequently the stator currents must be commutated electronically by means of power electronic converters. As the rotor rotates, each coil will have the role of a field coil for a portion of time and of a torque producing (armature) coil for the remainder of the time. Initially, [8], it was proposed to use square wave currents like in Fig. 1.8. However, trapezoidal field currents will produce continuous rotating MMF compared to step-rotating MMF produced by square wave currents which consequently reduces torque ripple, [9]. Boldea proposes a bipolar current waveform which doesn't go to zero during commutation between field and torque currents, while Lipo and Kamper propose current waveforms which do. They are presented in Fig. 1.9. In both cases commutation must be position triggered and proper control must be implemented. Phases are shifted by $\frac{\pi}{m}$ electrical, where m is the number of phases. Field and torque current will differ only in field weakening applications (i.e. reduction of field current), [10].

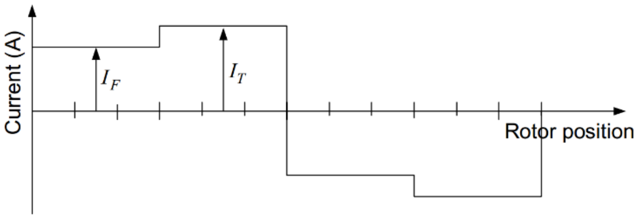
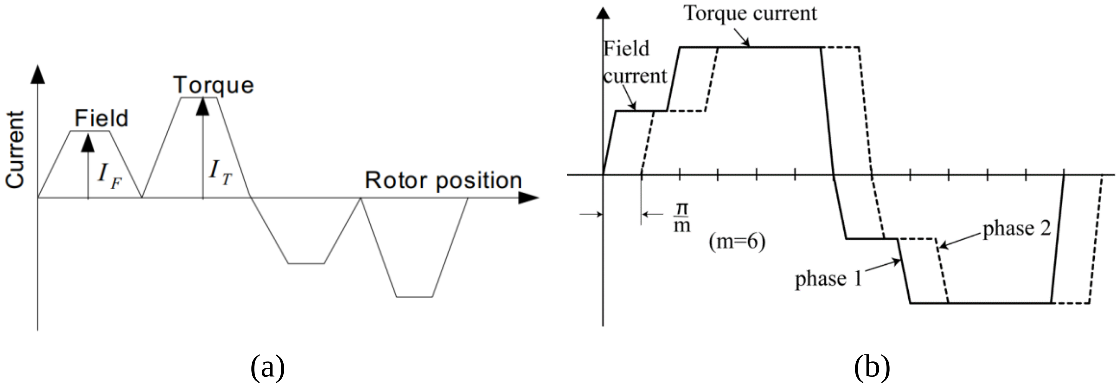


Fig. 1.8 Square wave current of six-phase BLDC reluctance machine



(a)

(b)

Fig. 1.9 Trapezoidal currents of six-phase BLDC reluctance machine proposed by (a) Lipo and (b) Boldea

As mentioned in the previous paragraph, for proper functioning of multi-phase BLDC reluctance machines, rotor saliency is required. Contrary to synchronous reluctance machines, in the case of multi-phase BLDC reluctance machines a high rotor saliency is required not for torque production but for reduced armature reaction, [11]. Nonetheless, nowadays there are many commercial and non-commercial ways to have high rotor saliency: segmental, flux barriers and axially laminated rotors. However, the most popular method is rotor with flux-barriers presented in Fig. 1.10. Another interesting option, although not yet commercially available, is to laminate the rotor of the machine in the axial direction. Axially laminated anisotropic (ALA) rotor is usually constructed using nested strips of steel separated with thin layer of the insulation [3]. Difference between radial (usual) and axial lamination is shown in Fig. 1.11. Axially laminated rotor before turning and grinding is presented in Fig. 1.12. Due to their high saliency ratio, ALA rotors have attracted a lot of attention over the years, however a simple and cheap way to mass produce them has not yet been found. It is reported that ALA rotor can achieve saliency up to 20 compared to segmental and rotor with flux barriers, which is up to 6-7 [12]. In [9] two different rotor structures (slitted and chamfered rotor) were explored and how they affect torque production, reduced armature reaction etc.

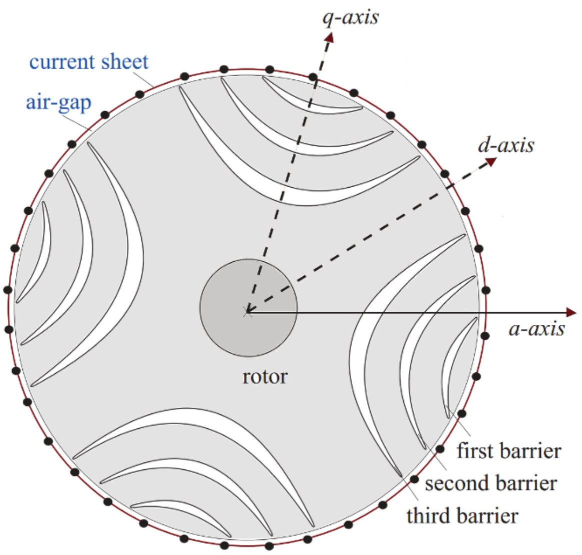


Fig. 1.10 Rotor with three flux barriers

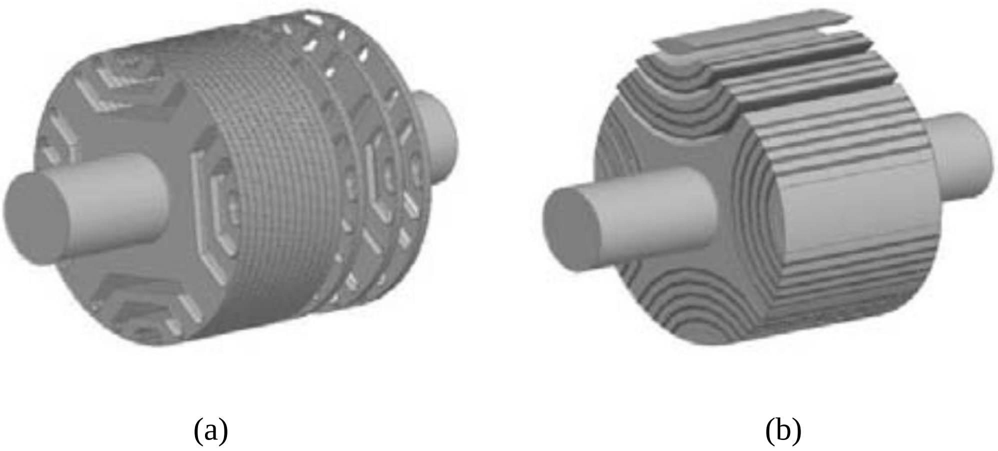


Fig. 1.11 Two different types of lamination: (a) radial (b) axial

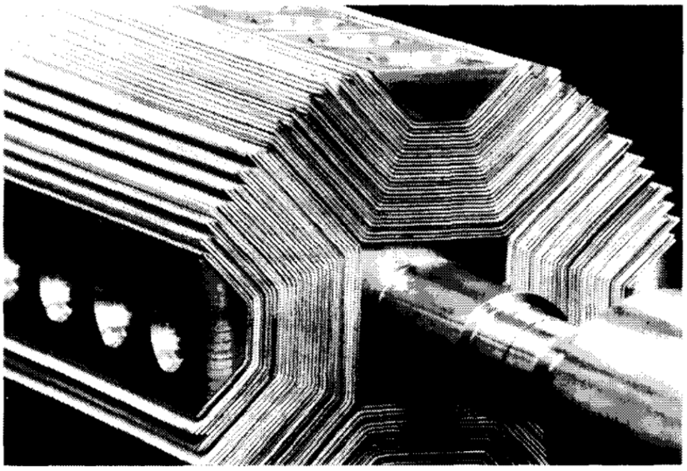


Fig. 1.12 Assembled 7.5kW ALA rotor before turning and grinding operations

Briefly summarized, it was shown that in DC machines, angle between stator and rotor flux is always 90°, by construction. Multiphase BLDC reluctance machine tries to manage the same thing with having both excitation (field) and armature (torque) winding at stator and to move magnetic poles to the rotor. This means that each conductor of the stator will act both as a field and torque winding, depending on the rotor position. For instance, let's consider machine in the Fig. 1.13. In this particular position, phases *a*, *b* and *c* are acting as torque coils whilst *d*, *e* and *f* as field. This pattern must be kept when the rotor rotates. For instance, if the rotor is rotating clock wise, phase *f* will become a torque coil and phase *c* a field coil.

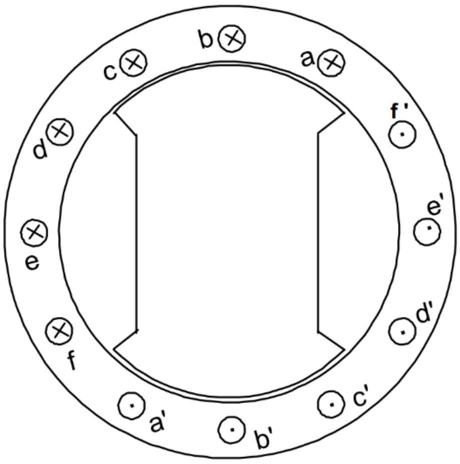


Fig. 1.13 Another configuration of 6 phase BLDC reluctance machine

Field coils are producing field in the direction of minimum reluctance (*d*-axis), while torque coils produce field in the direction of maximum reluctance (*q*-axis). For the machine and rotor position in Fig. 1.14, coil #3 and #4 are field currents while others are torque coil. In Fig. 1.15, fields of coils 3-3' and 4-4' are shown in the two pole model configuration where it is obvious that their sum is in the *d*-axis direction.

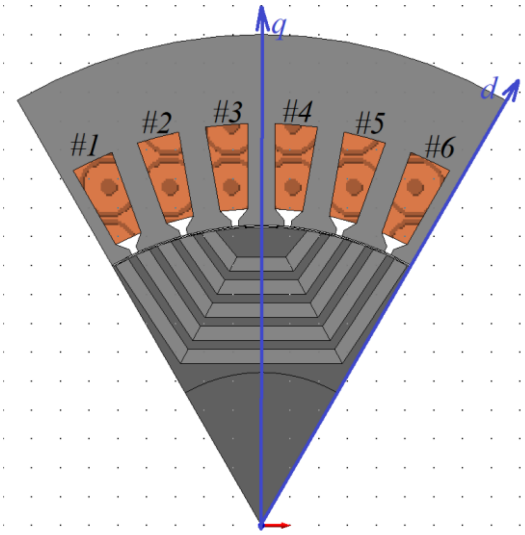


Fig. 1.14 Definition of *q* and *d* axis

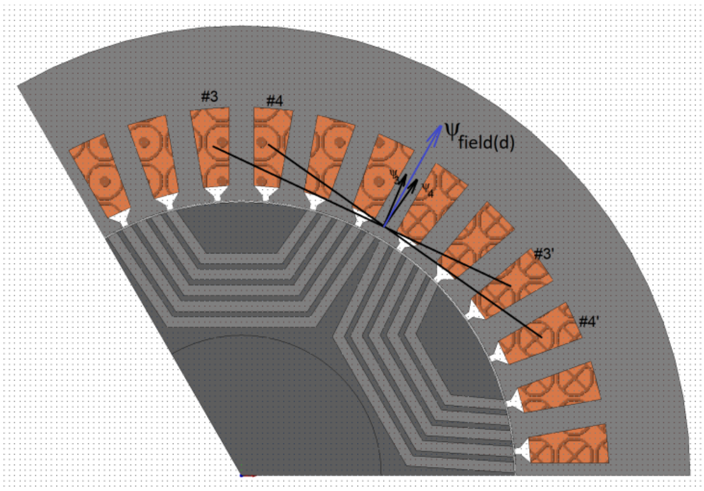


Fig. 1.15 Field flux in the direction of d-axis

1.3. Control Aspects of multi-phase BLDC Reluctance Machines

As it was mentioned in section 1.2, for proper functioning of a multi-phase BLDC reluctance machine a power converter is needed. Control of currents is done individually for each coil. Each phase can be controlled with a full-bridge converter shown in Fig. 1.16. However, the number of switches in this case is $4m$ where m is the number of phases. A more cost-effective solution, in terms of number of switches, is to use a half-bridge converter shown in Fig 1.17, where one end of the winding is connected to the mid-point of the leg while all other ends are connected to one mutual point called star point (Fig. 1.18). In this case a null conductor must exist. Current in the null conductor is oscillating from $-mI_{phase}$ and $+mI_{phase}$ and usually another half-bridge is present to accept that current or it is fed back to the DC link [4,10]. Therefore, using half-bridges for current waveform control, the number of switches is reduced from $4m$ to $2m+2$. The difference in number of switches is then $4m-(2m+2)=2m-2$. In a six-phase machine that is 10 switches. On the other hand, null leg of inverter must be oversized due to high null current and therefore transistors will be more expensive. Weh and Schroder in [4] proposed a way to reduce null currents with winding reversal and by keeping same MMF.

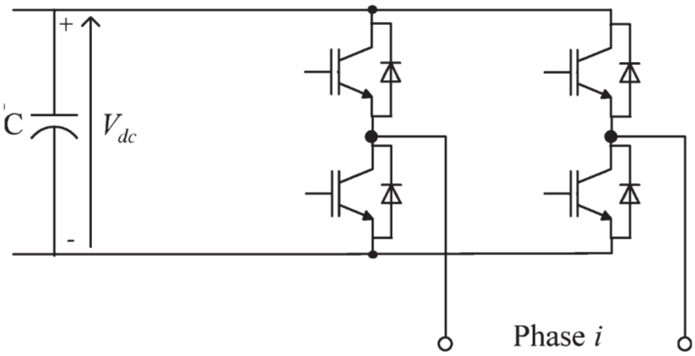


Fig. 1.16 Full-bridge converter

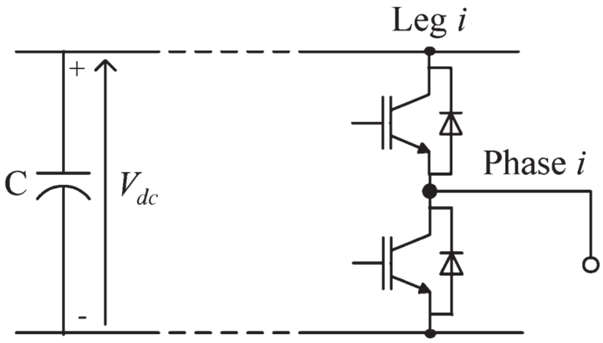


Fig. 1.17 Half-bridge converter

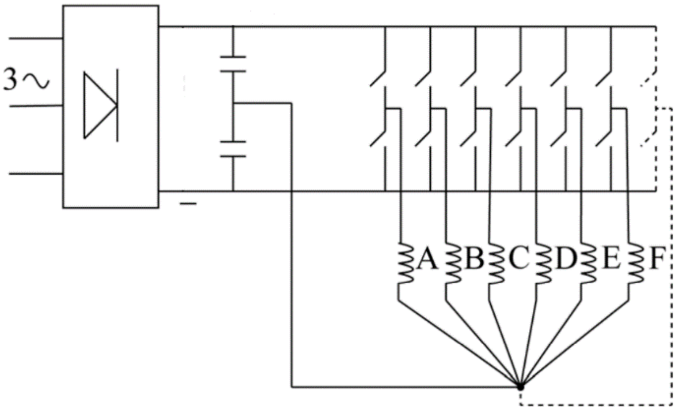


Fig. 1.18 Control of each phase with half-bridge converter and presence of null conductor

Full control scheme proposed by Law and Lipo [8] is presented in Fig 1.19. Each phase is controlled by full-bridge with IGBTs. However, they didn't specify control method used.

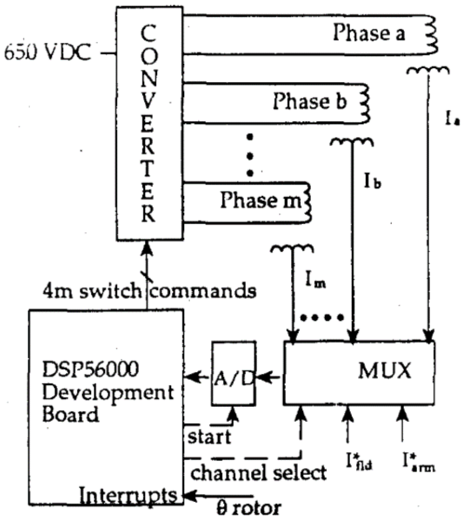


Figure 1.19 Drive configuration proposed by Lipo and Law

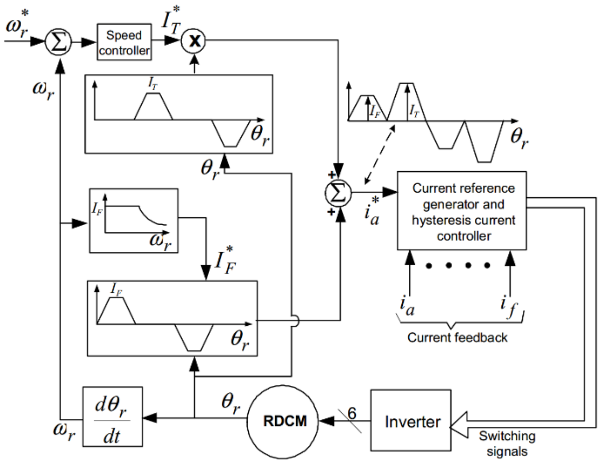


Figure 1.20 Block diagram of current control proposed by Kamper

Kamper described another control method in [9]. The drive configuration is the same as the one used by Lipo and Law shown in Fig. 1.19. The block diagram of the current control is presented in fig 1.20. The torque current is obtained from the speed controller, while the field current is set constant up to the field weakening region. For generation of current references, hysteresis control was implemented.

Boldea and his group have built a 5-phase BLDC reluctance machine and experimentally tested it and presented results in [11]. Each phase was controlled with a full-bridge inverter. On the other hand, in [13] each phase of a six-phase BLDC reluctance machine was controlled with a half-bridge inverter (Fig 1.21) For convenience, three industrial inverters were used. In [13], a PI controller was used for current control instead of hysteresis control since the latter one introduces high torque ripple and consequently high acoustic noise and vibrations.

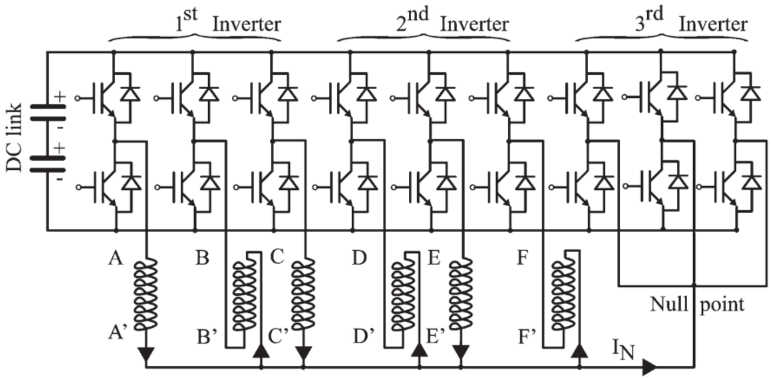


Fig. 1.21 Three industrial inverters scheme for control of six-phase machine

For modelling of the machine, $abcdef$ reference frame is used and set of equations (1.2) - (1.5). Park (dq) transformation is not used since phase model is closer to reality and nowadays computational power is not problem to solve equations (1.2) - (1.5).

$$\frac{d[\psi]}{dt} = [u] - [R][i] - [i]\frac{\partial[L(\theta)]}{\partial\theta} \tag{1.2}$$

$$[\psi] = [L(\theta)][i] \tag{1.3}$$

$$J\frac{d\Omega}{dt} = T_e - T_{load} - B\Omega \tag{1.4}$$

$$T_e = \frac{1}{2}[i]^T \frac{\partial[L(\theta)]}{\partial\theta}[i] \tag{1.5}$$

Clearly for the model of the machine parameters are needed. Parameter can be identified either experimentally or through finite element analysis. Some parameters which can be found experimentally are inductance (flux decay test), resistance and inertia. So far the most comprehensive and complete analysis of a multi-phase BLDC reluctance machine has been presented in [13]. Parameters of the already built machine were calculated and then verified through FE. Control system was developed with reduced number of switches and both motoring and generation mode were tested as well as field weakening operation. Also iron losses were calculated.

1.4. Advantages and Disadvantages

As commented in the introduction, the multi-phase BLDC reluctance machine concept was revived after high energy NdFeB price increase. However, it is not the only type which has been investigated. Synchronous reluctance and switched reluctance machines gained fairly high attention from scientific community and industry compared to multi-phase BLDC reluctance machine where limited number of papers on this topic exist. In spite of having dominant performance in some aspects, these two types have some disadvantages. For example, synchronous reluctance machines have poor power factor, torque density, constant power speed range (CPSR) unless saliency ratio is above 7 [3]. Thus, synchronous reluctance machine doesn't fully use inverter's kVA [14]. Despite high torque ripple, switched reluctance machines also don't use full possibilities of inverter [10]. Therefore, there are justified reasons to continue search for new types of the machines. Reasons why it is worth to pursue research of multi-phase BLDC reluctance machine are:

- Easy cooling
  Since there aren't conductors in the rotor almost all of the heat is generated in the stator and consequently it is easier to extract.
- Cost effective
  There is no need for high energy density and high cost PMs. Conventional three-phase am machine stators can be used.
- Reliability
  Existence of more phases allows operation even if one or more phases fail
- Required rated power per phase of the inverter is lowered

Applications in which this machine might find use in are railway traction, ship propulsion, wind generation, micro-hydro power plants etc.

Chapter 2

Finite Element Modelling of the Machine

The program which will be used for the analysis and simulation of the multi-phase BLDC reluctance machine is MagNet v7 from Infolytica corporation, while the drawing of the machine, the setting of all parameters and the post-processing will be done through Matlab scripts. This method has been chosen for numerous advantages. For instance, changes of various parameters of the machine can be done with only one or few line adjustments in the Matlab script, which will automatically be recognized by MagNet. This and other advantages will be demonstrated later.

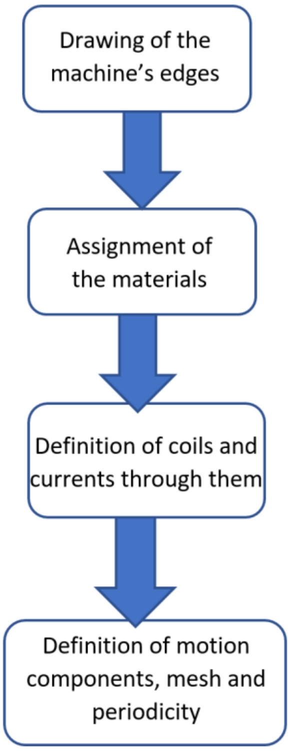

Drawing of the machine's edges
Assignment of the materials
Definition of coils and currents through them
Definition of motion components, mesh and periodicity

Fig. 2.1 Flowchart of the model preparation in MagNet

In the first part of this chapter, the scripting is explained. As it can be seen from Fig. 2.1 modelling starts with drawing of the machine's edges. In section 2.2. the drawing process is explained and missing geometric parameters are calculated while others are introduced. When all the edges are drawn, the single components are extruded and materials are assigned to them as explained in section 2.3. Here, axial lamination modeling is also explained in detail. After that, the coils and the associated currents are defined. For the model to function properly, it is also necessary to assign motion components, to select maximum element sizes for the mesh and to assign proper boundary conditions, as shown in sections 2.4 and 2.5. Each part of the modeling process is followed by a portion of the Matlab script which executes the above mentioned modelling procedures.

The main parameters of the machine are shown in table 2.1. Other geometric parameters are shown in the tables 2.2 and 2.3 respectively. The machine was built by prof. Boldea's research group at Polytechnic University of Timisoara and it is shown in Fig. 2.2.

Number of slots (N_s)	36
Number of phases (m)	6
Number of slots per pole per phase (q)	1
Stack length (L_s)	150mm
Number of poles (P)	6
Number of pair poles (p)	3
Rated field/torque current	12A/12A
Rated torque	36Nm
Rated speed	250rpm

Table 2.1. Main parameters of the machine

Slot opening (s_o)	1.2mm
Tooth width (t_w)	4.7mm
Stator inner diameter (sD_i)	104mm
Slot height 1 (sh_1)	14.8mm
Slot height 1 (sh_2)	1.8mm
Slot height 1 (sh_3)	1mm
Stator yoke height (h_{sy})	15.4mm

Table 2.2. Geometric parameters of the stator

Mechanical degree of one pole (θ_{pole})	$\theta_{pole} = \frac{360}{P} = 60°\text{mech}$
Air gap length (G_{ap})	0.35mm
Rotor inner diameter (rD_i)	53mm
Half of the rotor thickness	$\frac{sD_i}{2} - Gap - \frac{rD_i}{2} = 25.15\text{mm}$

Table 2.3. Some of the rotor's geometric parameters

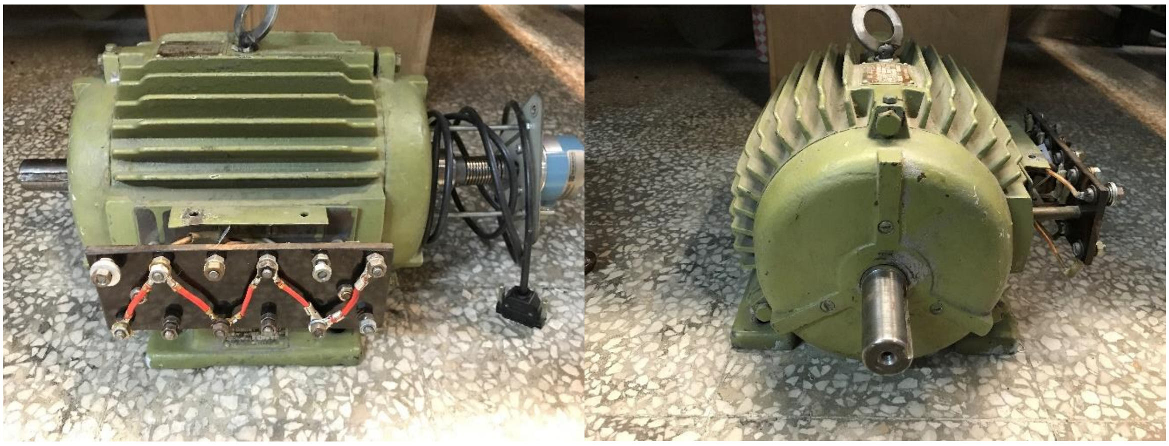


Fig. 2.2 Six-phase BLDC reluctance machine

2.1. Scripting and Connection of Matlab and MagNet

The classical way of model preparation in MagNet is interactively using mouse and keyboard. However, this method sometimes is not convenient when there are many repetitive tasks or when there are many modelling variations. In these cases, it is better to use scripting to automate those tasks. Scripting in MagNet is possible internally or externally [15]. External scripting means that it is possible for Matlab (or some other program) and MagNet to communicate via the ActiveX Automation Interface [16]. Advantages of external scripting are numerous and some of the are:

- Better GUI development
- Access to debugger
- More advanced data structures
- More advanced post-processing

The code which enables connection with the ActiveX link and opens a new MagNet document is shown below. The first line establishes the ActiveX link between MagNet and Matlab and creates a MN7 object. Later everything with MagNet will be referred to MN7. After that, MagNet is set to be visible and the universal constants are invoked. Finally, new document is opened and the objects *doc* and *view* are created. Also, it is convenient to set the default units and grid extent and spacing at the beginning of the code.

```
%MAGNET SETTINGS

MN7 = actxserver('MagNet-TrialEdition.Application'); %Open MagNet
set(MN7,'Visible','true'); %MagNet visible
Consts = invoke(MN7, 'getConstants'); %Call universal constants
invoke(MN7, 'newDocument'); %Open new document

Doc = invoke(MN7, 'getDocument');
view = invoke(Doc, 'getView');

%UNITS
%Setting of the default units
invoke(MN7, 'processCommand', 'getDocument.setDefaultLengthUnit("Millimeters")');
invoke(MN7, 'processCommand', 'getDocument.setDefaultTimeUnit("Milliseconds")');
invoke(MN7, 'processCommand', 'getDocument.setDefaultFrequencyUnit("Hertz")');
invoke(MN7, 'processCommand', 'getDocument.setDefaultTemperatureUnit("Celsius")');

%GRID
%Setting of grid extent and spacings
invoke(view,'showGrid','True');
invoke(view,'setGridExtent',-160,-160,160,160);
invoke(view,'setGridSpacings',5,5);
```

Most of the Matlab commands will start with *invoke*. The arguments of the invoke function are: object, command, arguments. In general, there are two types of commands:

- `invoke(MN7,'processCommand','Command in VBA/VBS');`

  The command above practically sends `'Command in VBA/VBS'` to MagNet which executes it. Some examples of commands in VBA/VBS are:

```
CALL getDocument().makeSimpleCoil(arguments)
CALL getDocument().getView().makeComponentInALine(arguments)
```

- `invoke(object in which function is executed,’function’,arguments);`

  For now, we have *Doc* and *view* objects. All functions which can be executed with these objects can be called in the way above without using ‘processCommand’.

- It is also possible to give commands in this way:

  ```
  MN7.set('Visible', 1); ( <=> set(MN7,'Visible','true');)
  MN7.invoke('processCommand',Command);
  ```

  where `Command` is previously defined string in which is written command in VBA/VBS

Some examples of useful commands are:

- For drawing arcs and lines

  ```
  invoke(view, 'newArc', 0, 0, XA1,YA1,XAn,YAn);
  invoke(view, 'newLine', XAn,YAn, XBn,YBn);
  ```

- For defining array

  ```
  invoke(MN7,'processCommand','REDIM NameComponent(1)');
  invoke(MN7,'processCommand','NameComponent(0)="Stator"');
  invoke(MN7,'processCommand','NameComponent(1)="Rotor"');
  ```

  It is important to mention that the values of the array must be string type, therefore if a number must be assigned to a member of the array, the Matlab function *num2str* must be used before. Example is shown below supposing ARRAY is already defined.

  ```
  value_str=num2str(value);
  invoke(MN7, 'processCommand', ['ARRAY(1)=',value_str]);
  ```

- For extracting data in post processing, for example the current flowing through coil

  ```
  MN7.invoke('processCommand', 'call
  getDocument.getSolution.getCurrentthroughCoil(problem, "Coil#1",
  magnitude1)');
  MN7.invoke('processCommand','call setVariant(0, magnitude1)');
  current1_f12_t12(k) = MN7.invoke('getVariant',0);
  ```

2.2. Parametric Drawing of the Machine via Matlab Scripts

Instead of drawing the machine directly in MagNet, it will be shown here how to draw it via a Matlab script. There are many repetitive procedures during the drawing procedure and they can be automated through scripting. However, the greatest advantage is probably the possibility of modifying any parameter of the machine, e.g., the stator yoke height, in a single file.

A device in MagNet is modelled using components [16]. For creation of the components, edges must be drawn in *construction slice*. Construction slice is a two-dimensional plane (xOy) in which edges are drawn. It is boarded with red rectangle in Fig. 2.3. Edges are usually drawn using simple circle, arc and line tools. Components are created from closed surfaces which are later extruded.

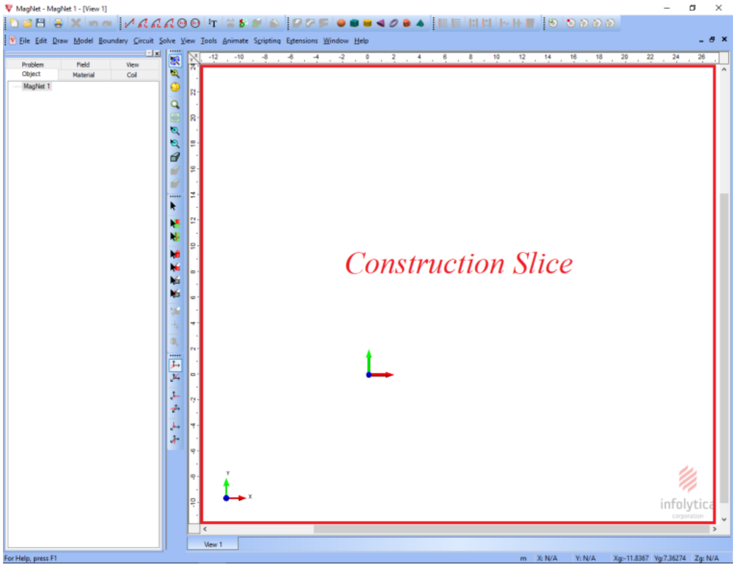

MagNet - MagNet 1 - [View 1]
File Edit Draw Model Boundary Circuit Solve View Tools Animate Scripting Extensions Window Help
Problem
Field
View
Object
Material
Coil
MagNet 1
Construction Slice
infolytica
View 1
For Help, press F1

Fig. 2.3 Construction slice

First, the stator will be drawn and then rotor of the machine. The stator parameters from table 2.2 are shown in Fig. 2.4. A missing parameter for the stator drawing is the stator outer diameter sD_o, which can be calculated according to (2.1) and is equal to 170mm.

$$sD_o = 2\left(\frac{sD_i}{2} + sh_1 + sh_2 + sh_3 + h_{sy}\right) \tag{2.1}$$

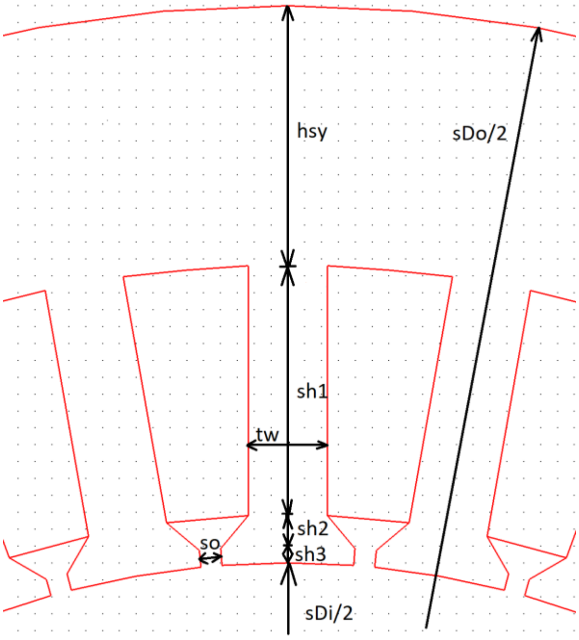


Fig. 2.4 Stator geometry and its parameters

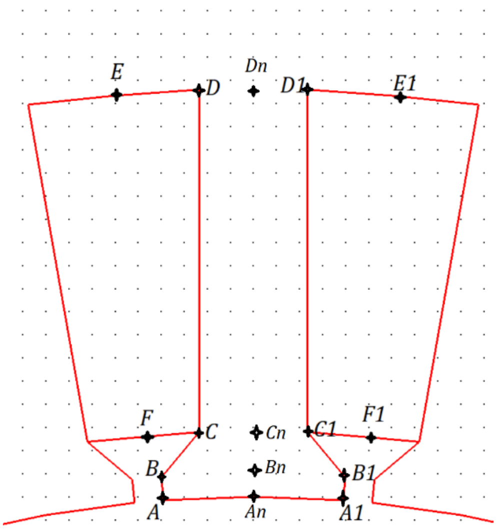


Fig. 2.5 Point enumeration for stator drawing

The necessary points for the stator drawing are shown in Fig. 2.5. Points $A, B, C, D, E, F, A_1, B_1, C_1, D_1, E_1, F_1$ define the tooth while points $A_n, B_n, C_n, D_n, E_n, F_n$ are used for drawing half of the tooth. Since only 1/6 of the machine is modeled due to symmetry, as explained later, we need to draw only $N_s/6$ teeth, i.e. six teeth. To this end, five full teeth will be drawn and two half teeth on each side, giving a total of six teeth and six slots. The code for stator drawing is presented below and the idea behind it is as follow. First a half of the tooth will be drawn as shown in Fig 2.6, followed by a *for* loop with (N_s/P-1) iterations to draw the other five teeth. In each iteration, points are connected with arcs and lines and are then prepared for another iteration, by rotating them for *-theta_slot*. At the end of the *for* loop, a half of the tooth is missed, Fig 2.7, so it has to be drawn together with an arc to close the stator back iron, as shown in Fig. 2.8. It is worth mentioning that, since the points $A_n, B_n, C_n, D_n, E_n, F_n$ don't participate in *for* loop, they are rotated by -60 degrees to complete the drawing. The code for this portion of the model is presented below. In the first lines, the parameters from the tables 2.1 and 2.2 are defined, since they will be used throughout script.

```
%MACHINE PARAMETERS

Ns=36; %Number of slots and teeth
theta_slot=360/Ns; %angle of the slot (angle between centres of two adjacent teeth)
slot_opening=1.2; %in mm
tooth_width=4.7; %in mm
m=6; %number of phases

sDi=104;     %stator inner diameter
sh1=14.8;    %slot height 1; from yoke
sh2=1.8;     %slot height 2
sh3=1;       % slot height 3 from air gap

hsy=15.4; %stator yoke height

sDo=2*(sDi/2+sh1+sh2+sh3+hsy); %stator outer diameter

arc_tooth=(sDi*pi*theta_slot)/360-slot_opening; %length of arc of the tooth; it is minus
       slot_opening because it is 2*(slot_opening/2)
theta_tooth=(arc_tooth*360)/(sDi*pi); %angle of the arc previously calculated; in
       degrees;
```

```
%rotor parameters

rDi=53; %rotor inner diameter
Gap=0.35; %airgap length
rDo=sDi-2*Gap; %rotor outer diameter
rotor_thickness=(rDo-rDi)/2;

% this parameters are depicted in the Fig. 2.8.
lamination_thickness=2.5; %lt denoted in the picture
insulation_thickness=1.5; %it denoted in the picture
distance=0.5; %distance of first point of lamination from axis of symetry (-60deg) (db)
distancerotor=1.5; %distance of first horizontal lamination from rotor (dr)
angle_defined=28; %another degree of freedom. angle of trapezoid.

%rotor
P=6; %number of poles
pp=P/2; %number of pole pairs
theta_pole=360/P;


%DRAWING

% A. Coordinate definition of points necessary for drawing
% These points are shown in Fig. 2.5

%PAIR (A,A1)
XA1=(sDi/2)*sind(theta_tooth/2);
YA1=(sDi/2)*cosd(theta_tooth/2);
XA=-(sDi/2)*sind(theta_tooth/2);
YA=YA1;

%PAIR (B,B1)
XB1=(sDi/2+sh3)*sind(theta_tooth/2);
YB1=(sDi/2+sh3)*cosd(theta_tooth/2);
XB=-XB1;
YB=YB1;

%PAIR (C,C1)
XC1=tooth_width/2;
YC1=sDi/2+sh2+sh3;
XC=-XC1;
YC=YC1;

%PAIR (D,D1)
XD1=tooth_width/2;
YD1=sDi/2+sh1+sh2+sh3;
XD=-XD1;
YD=YD1;

%PAIR (E,E1)
XE1=(sDi/2+sh3+sh2+sh1)*sind(theta_slot/2);
YE1=(sDi/2+sh3+sh2+sh1)*cosd(theta_slot/2);
XE=-XE1;
YE=YE1;

%PAIR (F,F1)
XF1=(sDi/2+sh3+sh2)*sind(theta_slot/2);
YF1=(sDi/2+sh3+sh2)*cosd(theta_slot/2);
XF=-XF1;
YF=YF1;

%PAIRS IN THE MIDDLE.
```

```
%Important only for the first and last iteration

XAn=0;
XBn=0;
XCn=0;
XDn=0;

YAn=sDi/2;
YBn=sDi/2+sh3;
YCn=YBn+sh2;
YDn=YCn+sh1;

%First iteration
%Connection of points with index 1 and n
%Only half of the tooth is drawn in Fig. 2.6.

invoke(view, 'newArc', 0, 0, XA1,YA1,XAn,YAn);
invoke(view, 'newLine', XAn,YAn, XBn,YBn);
invoke(view, 'newLine', XA1,YA1, XB1,YB1);
invoke(view, 'newLine',  XBn,YBn, XCn,YCn);
invoke(view, 'newLine',  XB1,YB1, XC1, YC1);
invoke(view, 'newLine',  XCn,YCn, XDn,YDn);
invoke(view, 'newLine',  XC1, YC1, XD1, YD1);
invoke(view, 'newArc', 0, 0, XE1,YE1,XD1,YD1);
invoke(view, 'newLine',  XC1, YC1, XF1, YF1);
invoke(view, 'newLine',  XDn,YDn, 0,sDo/2);

%Rotation of points for following iterations
[XA,YA]=rotatearoundorigin(XA,YA,-theta_slot);
[XB,YB]=rotatearoundorigin(XB,YB,-theta_slot);
[XC,YC]=rotatearoundorigin(XC,YC,-theta_slot);
[XD,YD]=rotatearoundorigin(XD,YD,-theta_slot);
[XE,YE]=rotatearoundorigin(XE,YE,-theta_slot);
[XF,YF]=rotatearoundorigin(XF,YF,-theta_slot);

[XA1,YA1]=rotatearoundorigin(XA1,YA1,-theta_slot);
[XB1,YB1]=rotatearoundorigin(XB1,YB1,-theta_slot);
[XC1,YC1]=rotatearoundorigin(XC1,YC1,-theta_slot);
[XD1,YD1]=rotatearoundorigin(XD1,YD1,-theta_slot);
[XE1,YE1]=rotatearoundorigin(XE1,YE1,-theta_slot);
[XF1,YF1]=rotatearoundorigin(XF1,YF1,-theta_slot);

%for the last iteration
%Points with n are used only for the last iteration

[XAn,YAn]=rotatearoundorigin(XAn,YAn,-60);
[XBn,YBn]=rotatearoundorigin(XBn,YBn,-60);
[XCn,YCn]=rotatearoundorigin(XCn,YCn,-60);
[XDn,YDn]=rotatearoundorigin(XDn,YDn,-60);

% loop for drawing rest of the (Ns/P)-1 teeth
%in each iteration

for i=1:((Ns/P)-1)
invoke(view, 'newArc', 0, 0, XA1,YA1,XA,YA);
invoke(view, 'newLine', XA,YA, XB,YB);
invoke(view, 'newLine', XA1,YA1, XB1,YB1);
invoke(view, 'newLine',  XB,YB, XC, YC);
invoke(view, 'newLine',  XB1,YB1, XC1, YC1);
invoke(view, 'newLine',  XC, YC, XD, YD);
invoke(view, 'newLine',  XC1, YC1, XD1, YD1);
invoke(view, 'newArc', 0, 0, XD, YD,XE,YE);
invoke(view, 'newArc', 0, 0, XE1,YE1,XD1,YD1);
invoke(view, 'newLine',  XC1, YC1, XF1, YF1);
invoke(view, 'newLine',  XC, YC, XF, YF);

%rotation before new iteration
[XA,YA]=rotatearoundorigin(XA,YA,-theta_slot);
[XB,YB]=rotatearoundorigin(XB,YB,-theta_slot);
[XC,YC]=rotatearoundorigin(XC,YC,-theta_slot);
[XD,YD]=rotatearoundorigin(XD,YD,-theta_slot);
[XE,YE]=rotatearoundorigin(XE,YE,-theta_slot);
[XF,YF]=rotatearoundorigin(XF,YF,-theta_slot);

[XA1,YA1]=rotatearoundorigin(XA1,YA1,-theta_slot);
[XB1,YB1]=rotatearoundorigin(XB1,YB1,-theta_slot);
[XC1,YC1]=rotatearoundorigin(XC1,YC1,-theta_slot);
[XD1,YD1]=rotatearoundorigin(XD1,YD1,-theta_slot);
[XE1,YE1]=rotatearoundorigin(XE1,YE1,-theta_slot);
[XF1,YF1]=rotatearoundorigin(XF1,YF1,-theta_slot);
end;

%after for loop we have structure shown in Fig 2.7.
%last portion i.e. iteration
%this part of the code is to draw missing half of the tooth

invoke(view, 'newArc', 0, 0, XAn,YAn,XA,YA);
invoke(view, 'newLine', XAn,YAn, XBn,YBn);
invoke(view, 'newLine', XA,YA, XB,YB);
invoke(view, 'newLine',  XBn,YBn, XCn,YCn);
invoke(view, 'newLine',  XB,YB, XC, YC);
invoke(view, 'newLine',  XCn,YCn, XDn,YDn);
invoke(view, 'newLine',  XC, YC, XD, YD);
invoke(view, 'newArc', 0, 0, XD,YD,XE,YE);
invoke(view, 'newLine',  XC, YC, XF, YF);
invoke(view, 'newLine',  XDn,YDn,(sDo/2)*cosd(30),(sDo/2)*sind(30));

%drawing of final arc and structure shown in Fig. 2.8.
invoke(view,'newArc',0,0,(sDo/2)*cosd(30),(sDo/2)*sind(30),0,sDo/2);

%Rotation of all edges for 30 degrees to obtain structure in Fig. 2.9.
invoke(MN7,'processCommand','Call getDocument().getView().selectAll(infoSetSelection,
       Array(infoSliceLine, infoSliceArc))');
invoke(view, 'rotateSelectedEdges', 0, 0,30,'False');
```

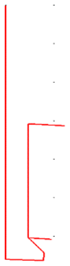


Fig. 2.6 Half of the tooth – first stage in stator drawing

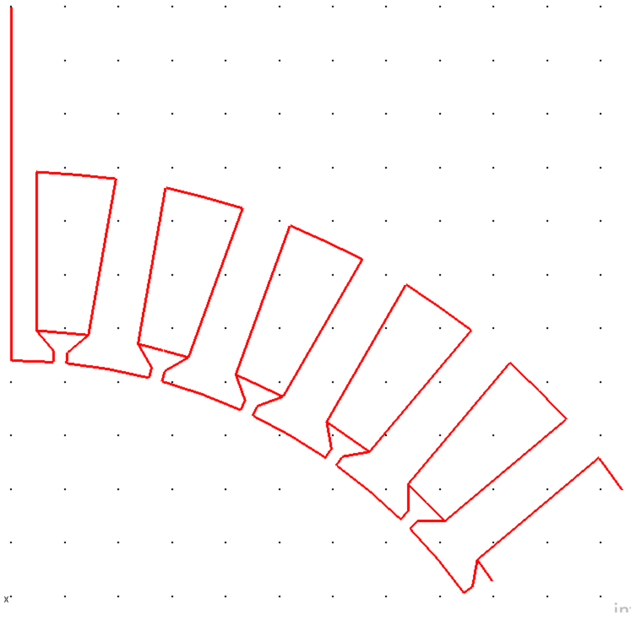


Fig 2.7 Stage after for loop

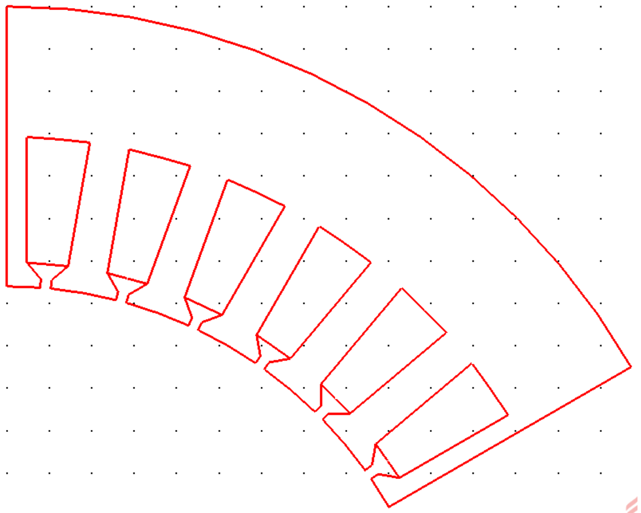


Fig 2.8 Completed stator of the machine

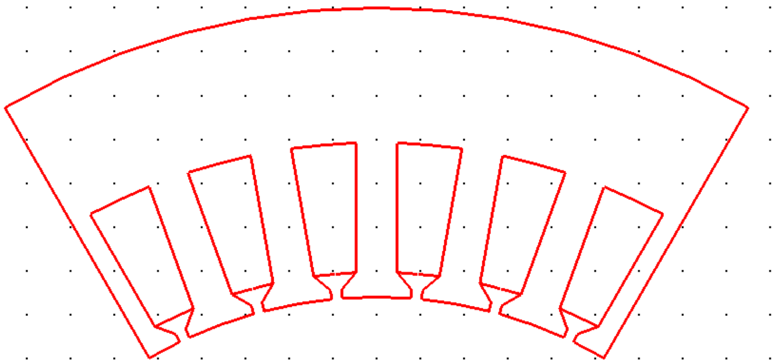


Fig 2.9 Final stator version. Rotated structure from Fig. 3.6. for 30 degrees

The final version of the stator with parameters from table 2.2. is shown in Fig. 2.9. However, the power of the script can also be demonstrated in the following example. If only the number of slots N_S is changed to 6 we get full machine geometry depicted in Fig. 2.10. If beside N_s, t_w is changed to 15mm and s_o to 50mm we get the geometry shown in Fig. 2.11., suitable for example for switched reluctance machines.

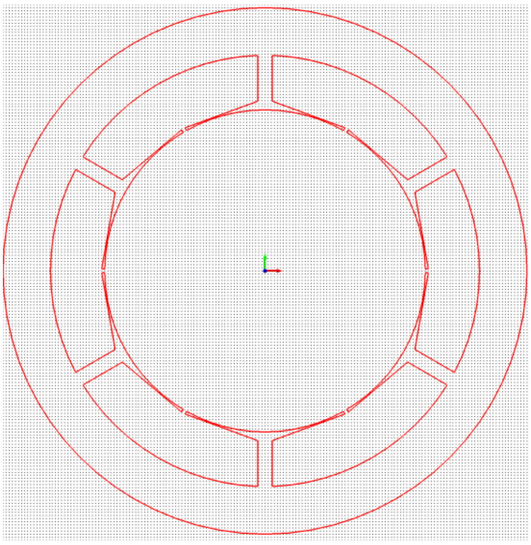


Fig. 2.10 Stator with same parameters as in table 2, but with $N_s = 6$.

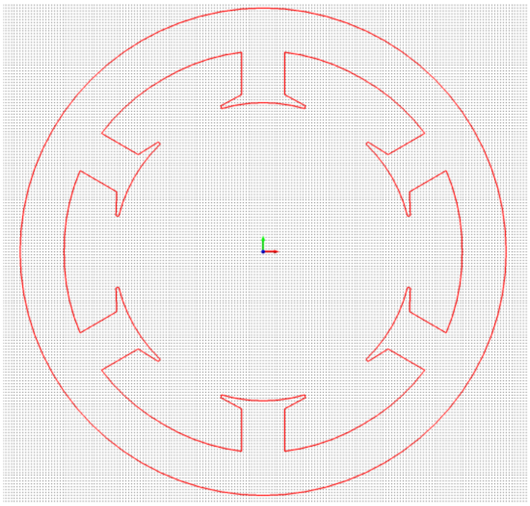


Fig. 2.11 Stator with $N_s = 6, t_w = 15\text{mm}$ and $s_o = 50\text{mm}$

Some of the rotor's parameters are shown in the table 2.3. However, with these parameters rotor cannot be drawn precisely. It is obvious that flux barriers and lamination ribs cannot be drawn unambiguously with given parameters. In order to draw the rotor's ribs, the new parameters d_b, l_t, i_t, d_r and *defined angle* are introduced; these are shown in Fig. 2.12. and their values are presented in table 2.4. By knowing these parameters, it is possible to unambiguously draw the ribs and the insulation barriers.

Defined angle	28
Distance from the rotor of the first lamination (d_r)	1.5mm
distance of the first point of the lamination from the border (d_b)	0.5mm
Lamination thickness (l_t)	2.5mm
Insulation thickness (i_t)	1.5mm

Table 2.4. Missing parameters (degrees of freedom) of the rotor's geometry

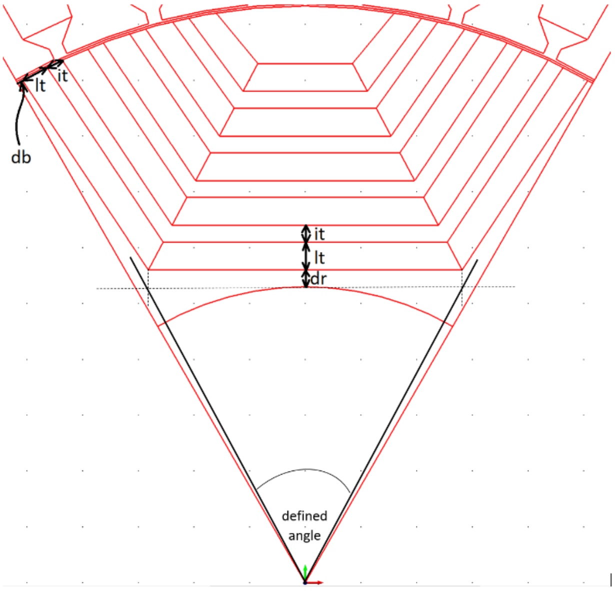


Fig. 2.12. Rotor geometry and its parameter

The rotor drawing starts by drawing the rotor outer arc, the inner rotor arc and the side lines. Furthermore, the airgap is split into two parts as required by the motion re-meshing, which will be explained in the section devoted to the meshing. Hence, another two arc commands are executed leading to the drawing shown in Fig. 2.13.

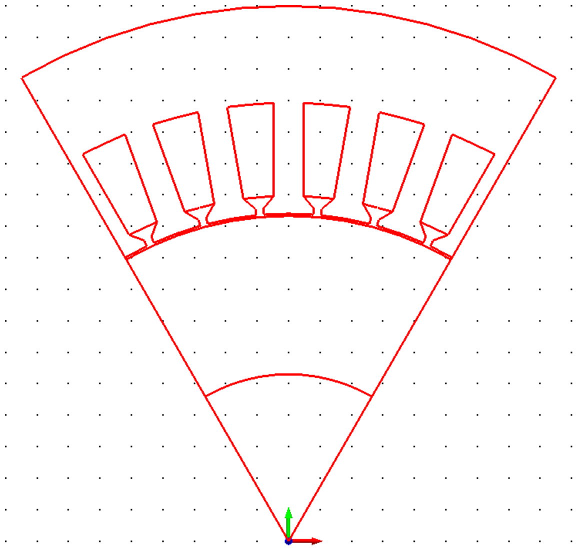


Fig. 2.13 Stage in rotor drawing

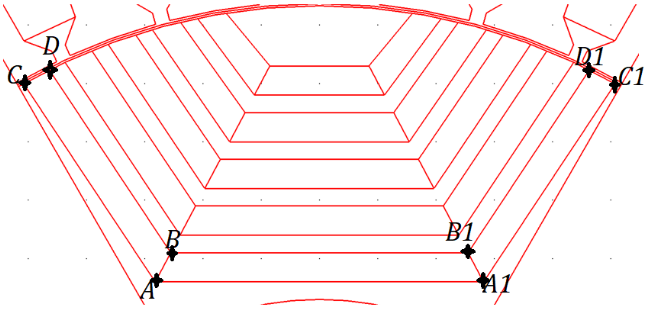


Fig. 2.14 Point enumeration for drawing of rotor’s ribs

In order to draw the rotor's ribs, coordinates of the points $A, B, C, D, A_1, B_1, C_1, D_1$ shown in Fig.2.14 are necessary. Three auxiliary variables *angle0*, *angle1* and *angle2* are created to obtain coordinates of points C and D. These are practically a measure of d_b, l_t and i_t in degrees. Point C is obtained by rotating point S by -*angle0*, and point D by -*angle0-angle1*. The coordinates of points A and B are obtained from the given data by using simple geometric relationship. When all the points are obtained, we can proceed with the ribs' drawing by using *for* loop. Similarly to the stator *for* loop, in each of every five iterations, first of all the connection with lines between points is established and then these are prepared for the following iteration. With this for loop, the drawing of the edges is finished. The final structure is presented in the Fig. 2.15.

```
% B. ROTOR
% Coordinate definition of points necessary for rotor drawing.
%These points are shown in Fig. 2.14

%drawing arcs of rotor and air gaps

invoke(view,'newArc',0,0,(rDo/2)*cosd(60),(rDo/2)*sind(60),-
       (rDo/2)*cosd(60),(rDo/2)*sind(60)); %outer rotor arc
invoke(view,'newArc',0,0,(rDi/2)*cosd(60),(rDi/2)*sind(60),-
       (rDi/2)*cosd(60),(rDi/2)*sind(60)); %inner rotor arc
invoke(view,'newLine',0,0,-(rDo/2+Gap)*cosd(60),(rDo/2+Gap)*sind(60)); %to close air gap
invoke(view,'newLine',0,0,(rDo/2+Gap)*cosd(60),(rDo/2+Gap)*sind(60)); to close air gap

%remesh
%air gap must be split in two parts

invoke(view,'newArc',0,0,((rDo+Gap)/2)*cosd(60),((rDo+Gap)/2)*sind(60),-
       ((rDo+Gap)/2)*cosd(60),((rDo+Gap)/2)*sind(60));
%another arc is added to separate stator air gap and parts of the slot which aren't Cu
invoke(view,'newArc',0,0,(sDi/2)*cosd(60),(sDi/2)*sind(60),-
       (sDi/2)*cosd(60),(sDi/2)*sind(60));

%start point (S,S1)

Xs=-(rDo/2)*sind(theta_pole/2);
Ys=(rDo/2)*cosd(theta_pole/2);
Xs1=-Xs;
Ys1=Ys;

%definition of auxiliary variables
%measurement of db, lt, it in angle

angle0=360*(distance/(rDo*pi));
angle1=360*(lamination_thickness/(rDo*pi));
angle2=360*(insulation_thickness/(rDo*pi));

%first ''rib''

XA=-(rDi/2)*tand(angle_defined);
YA=rDi/2+distancerotor;
XA1=-XA;
YA1=YA;

[XC,YC]=rotatearoundorigin(Xs,Ys,-angle0);
[XD,YD]=rotatearoundorigin(Xs,Ys,-angle0-angle1);
```

```
[XC1,YC1]=rotatearoundorigin(Xs1,Ys1,angle0);
[XD1,YD1]=rotatearoundorigin(Xs1,Ys1,angle0+angle1);

%vector normal to AC (important for direction of material)
%it will be used in assignment of material section
%it is convenient to calculate it at this moment when coordinates of A and C are known

dp=[XA-XC,YA-YC];
lengthdp=sqrt((XA-XC)^2+(YA*-YC)^2);
dp_unit=dp/lengthdp;
np_unit=[];
np_unit(1)=-dp_unit(2);
np_unit(2)=dp_unit(1);

np_unit_right=[];
np_unit_right(1)=-np_unit(1);
np_unit_right(2)=np_unit(2);


XB=XA+lamination_thickness*tand(angle_defined); %+because XA is negative
YB=YA+lamination_thickness;
XB1=-XB;
YB1=YB;

%we need this for assignment of the material

XA_copy=XA;
YA_copy=YA;
XA1_copy=XA1;
YA1_copy=YA1;
XB_copy=XB;
YB_copy=YB;
XB1_copy=XB1;
YB1_copy=YB1;
XC_copy=XC;
YC_copy=YC;
XC1_copy=XC1;
YC1_copy=YC1;
XD_copy=XD;
YD_copy=YD;
XD1_copy=XD1;
YD1_copy=YD1;
XE_copy=XE;
YE_copy=YE;
XE1_copy=XE1;
YE1_copy=YE1;

for i=1:5

invoke(view, 'newLine', XA,YA, XC,YC);
invoke(view, 'newLine', XD,YD, XB,YB);
invoke(view, 'newLine', XB,YB, XB1,YB1);
invoke(view, 'newLine', XB1,YB1, XD1,YD1);
invoke(view, 'newLine', XC1,YC1, XA1,YA1);
invoke(view, 'newLine', XA1,YA1, XA,YA);
invoke(view, 'newLine', XA1,YA1, XB1,YB1); %to distinguish right and central lamination
invoke(view, 'newLine', XA,YA, XB,YB); %to distinguish left and central lamination

[XC,YC]=rotatearoundorigin(XC,YC,-angle1-angle2);
[XD,YD]=rotatearoundorigin(XD,YD,-angle1-angle2);
[XC1,YC1]=rotatearoundorigin(XC1,YC1,angle1+angle2);
```

```
[XD1,YD1]=rotatearoundorigin(XD1,YD1,angle1+angle2);

XA=XA+(lamination_thickness+insulation_thickness)*tand(angle_defined);
YA=YA+lamination_thickness+insulation_thickness;
XA1=-XA;
YA1=YA;

XB=XB+(lamination_thickness+insulation_thickness)*tand(angle_defined);
YB=YB+lamination_thickness+insulation_thickness;
XB1=-XB;
YB1=YB;

End
```

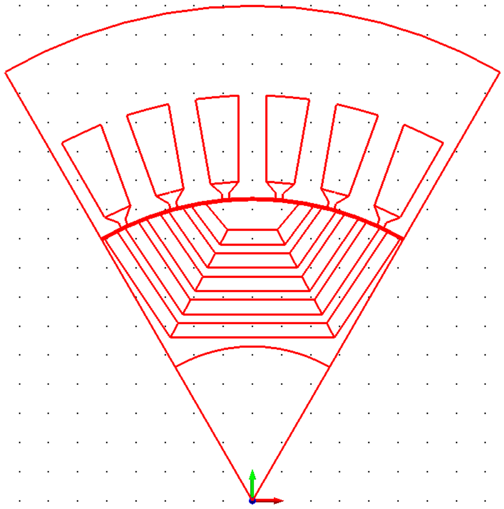
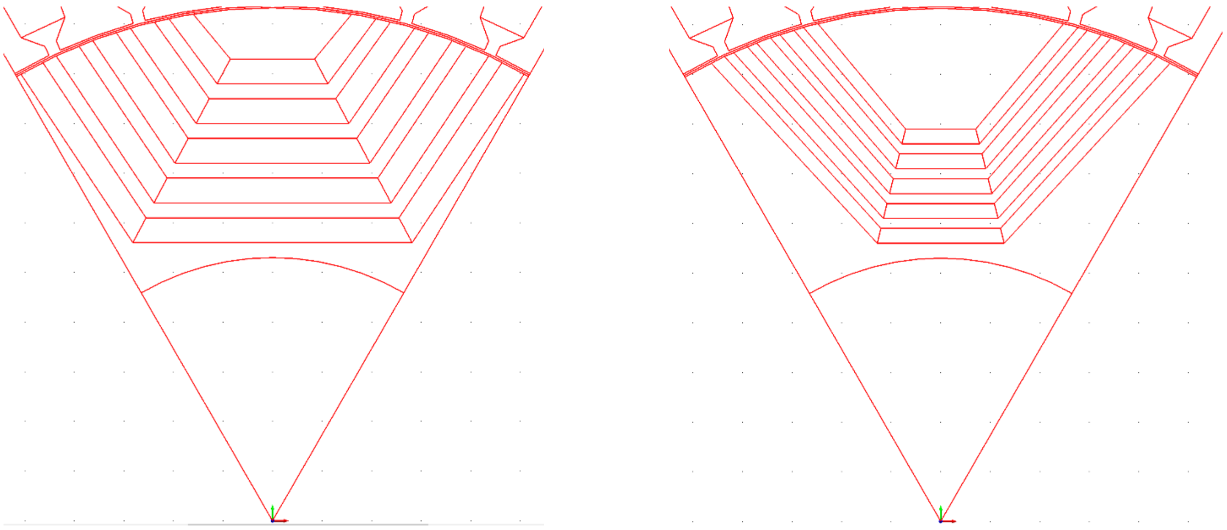


Fig. 2.15 Final structure

In Fig. 2.16 two configurations are shown which can be obtained by modifying some of the geometric parameters in the Matlab script. In Fig. 2.16(a) a configuration with parameters from tables 2.3. and 2.4. is shown. In Fig. 2.16(b) the lamination and insulation thickness are 1mm and 1.5mm and the defined angle is 14 degrees. Also d_b is changed to 3mm.



(a) (b)

Figure 2.16. Possible combinations of rotor lamination

It can be seen once again how the geometry of the model can be easily changed with the Matlab script. This effective and quick method of drawing the edges can be used to analyse how various geometric parameters affect performance of the machine.

2.3. Component Creation and Assignment of the Materials

Once the machine has been drawn in the construction slice, the next step is the creation of the machine components by extrusion, followed by the assignment of the materials. Extrusion and assignment of the materials are performed on construction slice surfaces, closed surfaces created by construction slice edges. Construction slice surfaces from which components have been made are: stator, air gap (rotor and stator), conductors, lamination (left, right and centre), insulation barrier, spider and rotor insulation. These are shown in Fig. 2.17 and Fig. 2.18. The reason for splitting the lamination into three parts will be explained in the following paragraph. Process of making the component is following. First construction slice surface is selected using *Select Construction Slice Surface* tool, and afterwards surface is extruded in direction normal to construction slice using *Make Component in A Line* tool. During sweeping, length and material are assigned to component. Thus, in Matlab script, which is presented below, `MakeComponentInALine()` function has three arguments: sweep length, name of the component and name of the assigned material. The assigned materials to surfaces are presented in table 2.5 and assigned length is stack length, which is 150mm. Model viewed from other angle so *z* (sweep) direction can be observed is shown in Fig. 2.19.

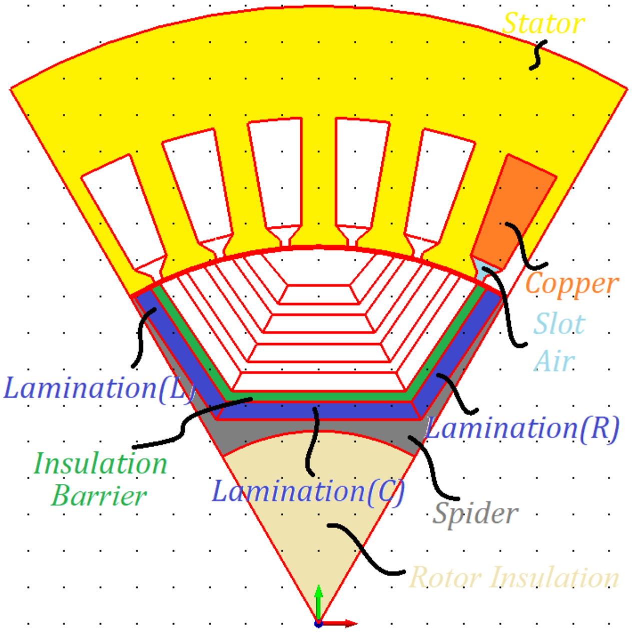


Fig. 2.17 Construction slice surfaces of the model

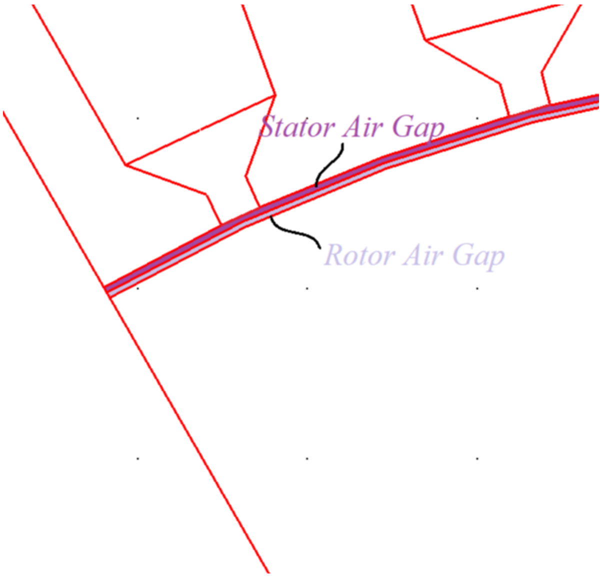


Fig. 2.18 Construction slice surfaces in the air gap

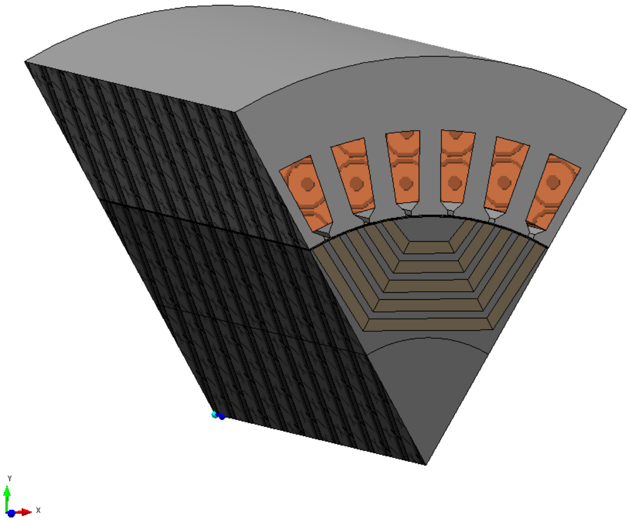


Fig. 2.19 Model when sweeping is applied

Part of the machine	Assigned Material
Stator	M300-35A
Conductor	Copper
Air Gaps (both rotor and stator)	Air
Insulation Barriers, Spider and Rotor Insulation	Stainless steel
Lamination	User created material

Table 2.5. Materials assigned to various parts of the machine

As mentioned in Chapter 1, the rotor has flux barriers and axially laminated material. The flux barriers are filled with non-magnetic insulating material, while the flux paths are built from axially laminated magnetic steel. Since axially laminated material has a higher reluctance compared to the parallel direction, a non-linear anisotropic material seems as a viable option to be used. Hence, a non-linear anisotropic user defined material was created

(Lamination_parallel_normal-1). Since the material is anisotropic its properties depend of directions x, y and z. This property can be described with equation (2.2)

$$\begin{bmatrix} Bx \\ By \\ Bz \end{bmatrix} = M \begin{bmatrix} Hx \\ Hy \\ Hz \end{bmatrix} \tag{2.2}$$

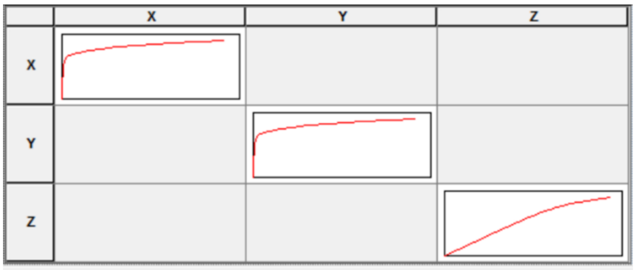


Fig. 2.20. Shape of matrix *M* from equation (2.2)

Since the material is non-linear, the *M* matrix on main diagonal doesn't have constant values but it has three *B-H* curves as shown in Fig. 2.20. In MagNet's 2D simulations, the coordinate system of the material is (xx, yy, zz) and when defining the *M* matrix, the xx and yy permeabilities must be the same. When assigning this material to the component, it is necessary to indicate the material direction, i.e., how the zz axis of the material is assigned in relation to the global coordinate system (x, y, z). Since all of the laminations in the centre, left and right part of the rotor have the same normal vectors, respectively n_1, n_2 and n_3,that is the reason why lamination was split in three parts, Fig. 2.21. For material direction type it is selected to be uniform direction, which means that direction will follow the component during its rotation, as shown in Fig. 2.22.

Vectors n_2 and n_3 are calculated during the rotor drawing procedure. Vector n_2 is perpendicular to line AC, shown in Fig. 2.14, therefore the problem is to find normal vector tor AC, since the coordinates of A and C are known. Points $A(x_A, y_A)$ and $C(x_C, y_C)$ define vector $AC = (-x_C, y_A - y_C)$. Vectors AC and n_2 are orthogonal if $AC * n_2 = 0$, where $*$ is the scalar product of two vectors. Hence, if vector n_2 has cooridnates $-(x_A - x_C)$ and $y_A - y_C$ the scalar product will be zero. Vector n_3 has same y coordinate as n_2 and opposite x coordinate. All calculations are performed with unit vectors. Vector n_1 has (0,1) coordinates since it is the unit vector of the y-axis.

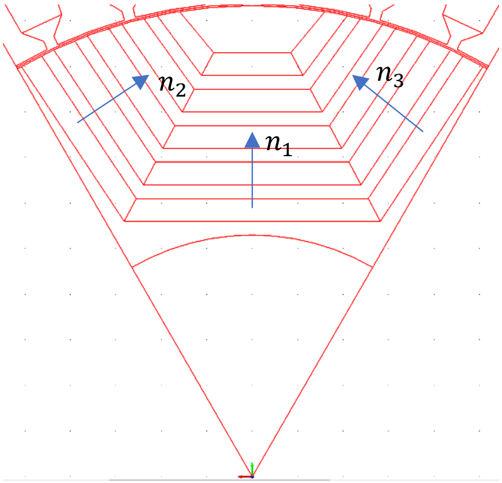


Fig. 2.21 Three different normal vectors which define rotor lamination

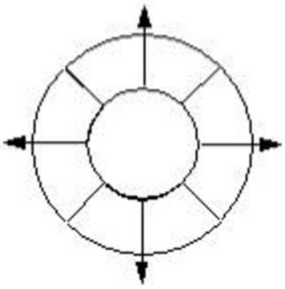


Fig. 2.22 Uniform direction type

According to MagNet's Online Help [16], for anisotropic materials and laminated materials the following procedure was used to obtain the non-linear anisotropic *B-H* curves necessary for the *M* matrix.

$$\bar{B}_{\parallel}(H_{\parallel}) = pB(H_{\parallel}) + (1-p)\mu_0 H_{\parallel} \tag{2.3}$$

$$\bar{H}_{\perp}(B_{\perp}) = pH(B_{\perp}) + (1-p)\frac{B_{\perp}}{\mu_0} \ldots (2.4) \tag{2.4}$$

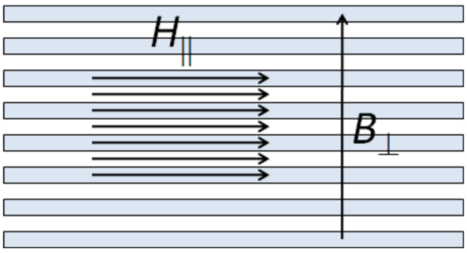


Fig. 2.23. Small piece of lamination

Formulas (2.3) and (2.4) are obtained by considering a small lamination volume in which it can be considered that $H_{\parallel}$ and $B_{\perp}$ are constant. Since $H_{\parallel}$ is going through the magnetic and non-magnetic material (the stacking factor indicates in which ratio) we can find the average value of $B_{\parallel}$ with the previous formula. First part in (2.3) is B in magnetic material, calculated through the $B-H$ curve and the second part is B in the insulation of the lamination (linear non-magnetic material). The same idea can be used for $H_{\perp}$. The lamination material is created from the M300-35A $B-H$ curve using a Microsoft Excel sheet provided form Infolytica and using a stacking factor of 0.97. Different $B-H$ curves can be seen in Fig. 2.24. Obviously, the preferred path for the field is parallel.

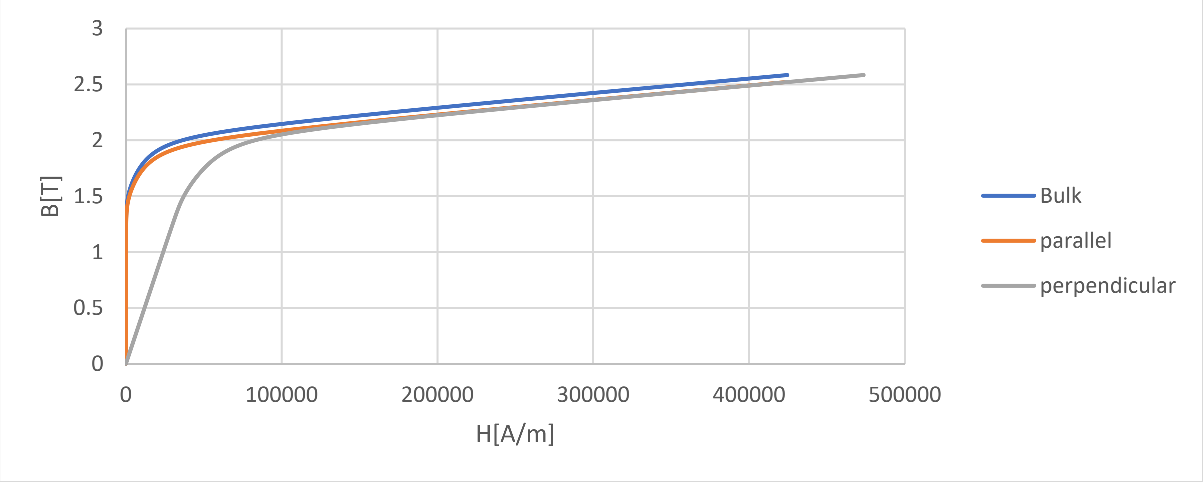


Fig. 2.24 $B-H$ curves of bulk M300-35A, parallel and normal direction

```
%ASSIGNMENT OF THE MATERIALS

%Stator

invoke(view,'selectAt',0,sDi/2+sh3,get(Consts,'infoSetSelection'));
invoke(MN7,'processCommand','REDIM NameComponent(0)');
invoke(MN7,'processCommand','NameComponent(0)="Stator"');
invoke(MN7,'processCommand',['CALL
       getDocument.makeComponentInALine(150,NameComponent,"Name=M300-35A")']);
%next line must be uncommented if linear material is desired and previous one commented
%invoke(MN7,'processCommand',['CALL
       getDocument.makeComponentInALine(150,NameComponent,"Name=MU3: Relative
       permeability 1000")']);


%Air Gap (Rotor and Stator)

invoke(view,'selectAt',0,rDo/2+(Gap-0.1)/2,get(Consts,'infoSetSelection'),
       get(Consts,'infoSliceSurface'));
invoke(MN7,'processCommand','REDIM NameComponent(0)');
invoke(MN7,'processCommand',['NameComponent(0)="RotorAirGap"']);
%MakeComponentInALine arguments are length, name of the component and material
invoke(MN7,'processCommand',['CALL
       getDocument.makeComponentInALine(150,NameComponent,"Name=AIR")']);

invoke(view,'selectAt',0,sDi/2-
       0.05,get(Consts,'infoSetSelection'),get(Consts,'infoSliceSurface'));
invoke(MN7,'processCommand','REDIM NameComponent(0)');
invoke(MN7,'processCommand',['NameComponent(0)="StatorAirGap"']);
invoke(MN7,'processCommand',['CALL
       getDocument.makeComponentInALine(150,NameComponent,"Name=AIR")']);

%Rotor
% for selection of central part of the rib

Xm=0;
Ym=rDi/2+distancerotor+lamination_thickness/2;
```

```
Ym_begin=rDi/2+distancerotor+lamination_thickness/2;

for i=1:5

k=1;

%central part of the rib

invoke(view,'selectAt',Xm,Ym,get(Consts,'infoSetSelection'));
invoke(MN7,'processCommand','REDIM NameComponent(0)');
invoke(MN7,'processCommand',['NameComponent(0)="Lamination',num2str(i),'of pole',num2str(k),'"']);
invoke(MN7,'processCommand',['CALL getDocument.makeComponentInALine(150,NameComponent,"Name=Lamination_parallel_normal-1;Type=Uniform;Direction=[0,1,0]")']);

%left part of the rib

invoke(view,'selectAt',XA_copy,YA_copy+0.05,get(Consts,'infoSetSelection'));
invoke(MN7,'processCommand','REDIM NameComponent(0)');
invoke(MN7,'processCommand',['NameComponent(0)="Laminationleftup',num2str(i),'of pole',num2str(k),'"']);

%in xstr and ystr will be put coordinates of normal vector
%this vector was calculated in drawing part
xstr=num2str(np_unit(1));
ystr=num2str(np_unit(2));

invoke(MN7,'processCommand',['CALL getDocument.makeComponentInALine(150,NameComponent,"Name=Lamination_parallel_normal-1;Type=Uniform;Direction=[',xstr,',',ystr,',0]")']);

%right part of the rib

invoke(view,'selectAt',XA1_copy,YA1_copy+0.05,get(Consts,'infoSetSelection'));
invoke(MN7,'processCommand','REDIM NameComponent(0)');
invoke(MN7,'processCommand',['NameComponent(0)="Laminationrightup',num2str(i),'of pole',num2str(k),'"']);

xstr=num2str(np_unit_right(1));
ystr=num2str(np_unit_right(2));

invoke(MN7,'processCommand',['CALL getDocument.makeComponentInALine(150,NameComponent,"Name=Lamination_parallel_normal-1;Type=Uniform;Direction=[',xstr,',',ystr,',0]")']);

%prepare selection points for the next iteration
%for central rib

Ym=Ym_begin+i*(lamination_thickness+insulation_thickness);
Xm=0;

XA_copy=XA_copy+(lamination_thickness+insulation_thickness)*tand(angle_defined);
YA_copy=YA_copy+lamination_thickness+insulation_thickness;
XA1_copy=-XA_copy;
YA1_copy=YA_copy;

end

%Spider
```

```
invoke(view,'selectAt',0,rDi/2+distancerotor/2,get(Consts,'infoSetSelection'));
invoke(MN7,'processCommand','REDIM NameComponent(0)');
invoke(MN7,'processCommand',['NameComponent(0)="Spider"']);
invoke(MN7,'processCommand',['CALL
       getDocument.makeComponentInALine(150,NameComponent,"Name=304 Stainless steel")']);

%Insulation
% point for insulation selection

Xm=0;
Ym_begin=rDi/2+distancerotor+lamination_thickness+insulation_thickness/2;
Ym=rDi/2+distancerotor+lamination_thickness+insulation_thickness/2;

for i=1:5

k=1;

invoke(view,'selectAt',Xm,Ym,get(Consts,'infoSetSelection'));
invoke(MN7,'processCommand','REDIM NameComponent(0)');
invoke(MN7,'processCommand',['NameComponent(0)="Insulation',num2str(i),'of
       pole',num2str(k),'"']);
invoke(MN7,'processCommand',['CALL
       getDocument.makeComponentInALine(150,NameComponent,"Name=304 Stainless steel")']);

%prepare selection point for the next iteration

Ym=Ym_begin+i*(lamination_thickness+insulation_thickness);
Xm=0;

end

%copper & air of the slot

XF1=(sDi/2+sh3+sh2+sh1/2)*cosd(60+theta_slot/2);
YF1=(sDi/2+sh3+sh2+sh1/2)*sind(60+theta_slot/2);
XF2=(sDi/2+sh3)*cosd(60+theta_slot/2);
YF2=(sDi/2+sh3)*sind(60+theta_slot/2);

for i=1:Ns/P

invoke(view,'selectAt',XF1,YF1,get(Consts,'infoSetSelection'));
invoke(MN7,'processCommand','REDIM NameComponent(0)');
invoke(MN7,'processCommand',['NameComponent(0)="Copper',num2str(i),'"']);
invoke(MN7,'processCommand',['CALL
       getDocument.makeComponentInALine(150,NameComponent,"Name=Copper: 5.77e7
       Siemens/meter")']);

%preparation for next iteration
%rotation of (XF1,YF1) to adjacent slot

[XF1,YF1]=rotatearoundorigin(XF1,YF1,theta_slot);

invoke(view,'selectAt',XF2,YF2,get(Consts,'infoSetSelection'));
invoke(MN7,'processCommand','REDIM NameComponent(0)');
invoke(MN7,'processCommand',['NameComponent(0)="AirSlot',num2str(i),'"']);
invoke(MN7,'processCommand',['CALL
       getDocument.makeComponentInALine(150,NameComponent,"Name=AIR")']);

%preparation for next iteration
%rotation of (XF2,YF2) to adjacent air slot
```

```
[XF2,YF2]=rotatearoundorigin(XF2,YF2,theta_slot);

end

%Rotor Insulation
invoke(view,'selectAt',0,rDi/2-2,get(Consts,'infoSetSelection'));
invoke(MN7,'processCommand','REDIM NameComponent(0)');
invoke(MN7,'processCommand',['NameComponent(0)="Rotor Insulation"']);
invoke(MN7,'processCommand',['CALL
       getDocument.makeComponentInALine(150,NameComponent,"Name=304 Stainless steel")']);
```

2.4. Coils, Currents and Motion Components

The coils in the machine are concentrated full-pitched windings. Each of the six phases has 28 turns. These are set to be current driven and stranded. Stranded coils assume that the turns of each coil are electrically isolated, uniformly distributed over cross section, connected in series and carry uniform current density over its cross section. As mentioned before, commutation of current is dependent on rotor position, therefore speed of the rotation of the machine is needed to define the currents. In the following part of the Matlab script, the speed of the machine and the relevant parameters for the simulation such as the time step and the end time are defined.

```
speed_rpm=-250;
speed=(speed_rpm*360/60); % MagNet speed input is in deg mech/s
time_start=0;
degree_electrical_end=180; %ending moment in el degrees
degree_mechanical_end=degree_electrical_end/pp;
time_end=((degree_mechanical_end/abs(speed))*1000); %ending moment in ms
angular_resolution_el=0.5;
time_step=(angular_resolution_el/(pp*abs(speed)))*1000; %resolution in ms
```

In the code above the speed was defined, together with other simulation parameters which are speed dependant, such as the ending time and the time step. In fact, the ending time is calculated in accordance with how many electrical degrees are desired for a simulation. Also, the user defines the angular resolution in electrical degrees which is later translated to time steps in *ms*. This parameter directly affects time required for a complete simulation.

```
rise_time_mechanical=2;
rise_time_electrical=rise_time_mechanical*pp;
rise_time=(rise_time_electrical/angular_resolution_el)*(time_step/1000);
fall_time=rise_time;
period=(120/abs(speed));
p12=period/12;

i_field=12;
i_torque=0;
```

The parameters required for the definition of the current waveform are the period, 1/12 of the period, rise and fall time and field (I_F) and torque (I_T) currents level. As discussed in Chapter 1, currents will be trapezoidal (to reduce the ripple) pulsed from $-I_F$ to I_F ($I_F = I_T$ except in field weakening). The time required for commutation is set to be 2 mechanical degrees (which can be changed in the script; rise and fall time), and it fulfils condition to be less than 10 degrees, i.e., the value of the slot pitch [11]. Currents of each phase are shifted by $\frac{\pi}{6}$ electrical (1/12 of the electrical period). At each instant of time some coils behave as torque windings and other as field windings. This pattern must be rotated as the rotor rotates. At the beginning of the simulation, currents of coil #3 and coil #4 are set to be field currents.

For the position of the rotor shown in figure 2.25, the currents have a certain pattern to provide the correct d and q axis fluxes, ψ_d and ψ_q. As the rotor rotates, ψ_d and ψ_q must also rotate, therefore the current pattern must be exchanged among the phases as the rotor rotates. At the beginning currents of coils 1 to 4 are negative and 5 and 6 are positive. Since the rotor is rotating clock-wise coil #5 will be the first which will change the pattern (positive to negative) according to the speed, which can be seen in Fig. 2.26. – Fig. 2.28. However, commutation will start after 5 mechanical degrees when the q axis is in the middle of slot #4. Also, it can be seen from Fig. 2.26. – Fig. 2.28 that commutation for coil #5 starts at the same time as for coil #1 and it ends at the same time as well. It is very important to start commutation at correct instant in order to obtain correct machine operation.

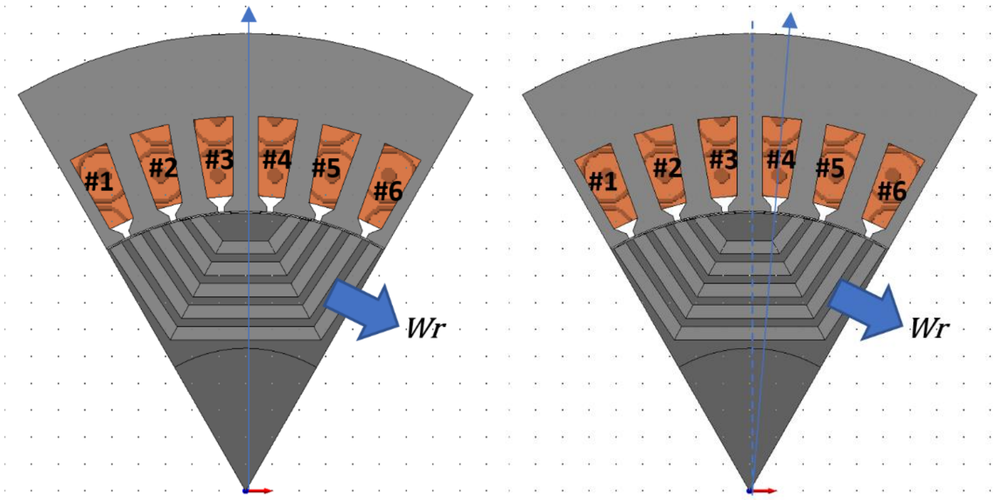


Fig. 2.25 Coil numeration

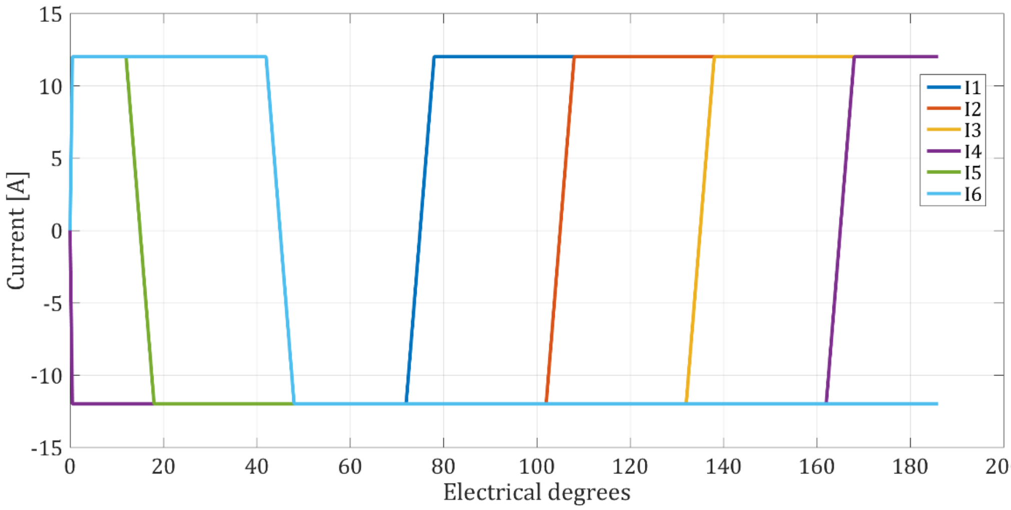


Fig. 2.26 Current waveform in rated conditions $I_F = I_T = 12A$

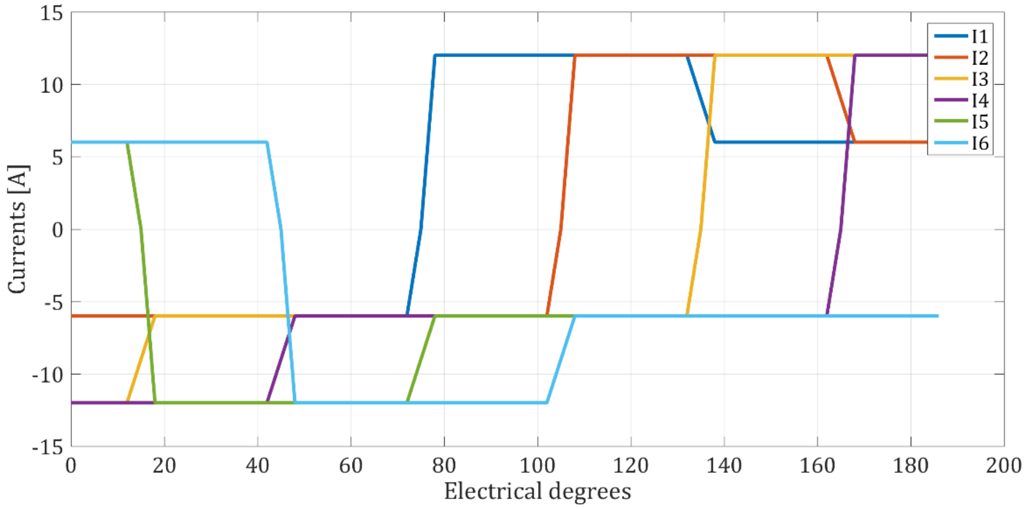


Fig. 2.27 Current waveform with $I_F = 12A$, $I_T = 6A$

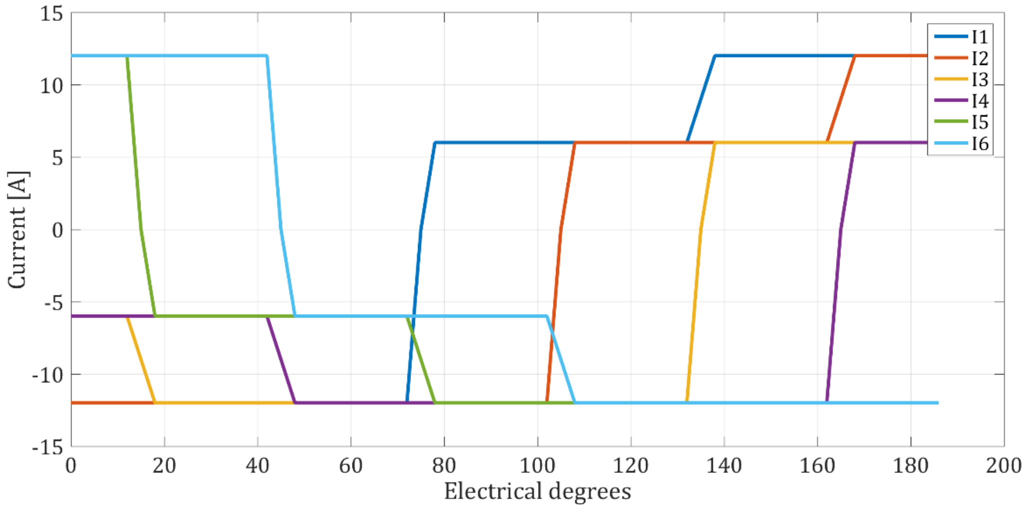


Fig. 2.28 Current waveform with $I_F = 6A$, $I_T = 12A$

Among possible waveforms, it has been decided to select piece wise linear type (PWL). For the PWL waveform, a vector of time (t) must be defined together with a vector of values (v). In the script, first, points of auxiliary current which start at 0 *ms* and goes to $-I_F$ will be defined (Fig. 2.29). Then this current will be shifted first by 5 mechanical degrees (15 electrical degrees) in order to obtain current of phase 5 (Fig. 2.30). After that currents will be shifted for $\frac{\pi}{6}$ to obtain other currents. Part of code which defines currents is shown below. Only script for current 5 will be explained since pattern for other currents is completely the same. Only *t_pwl_help* variable is shifted specific amount of *p12* times. Time instants for definition of auxiliary current are presented in the Matlab script and they are shown in Fig. 2.31.

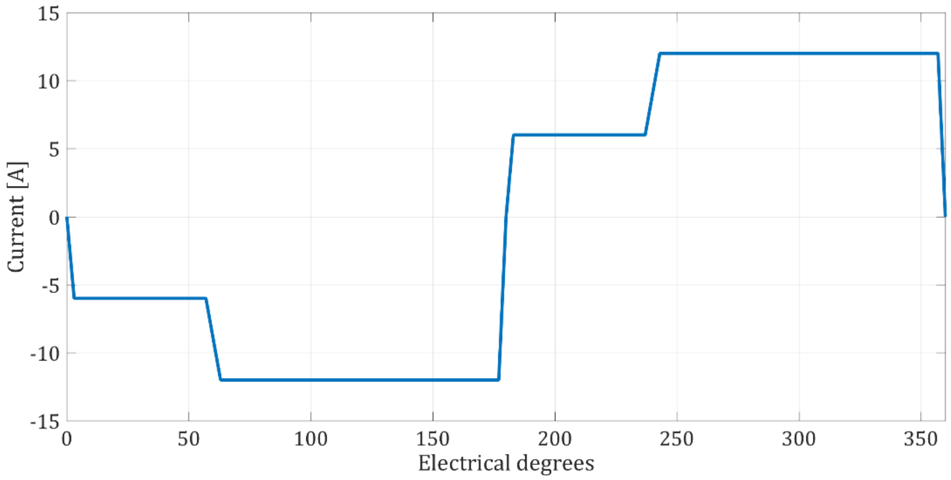


Fig. 2.29 Auxiliary current from which all other currents are obtained

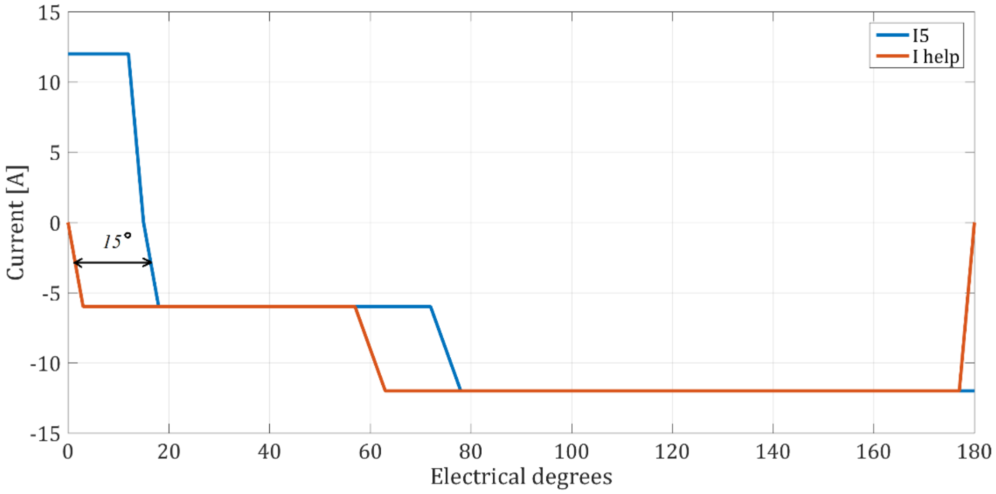


Fig. 2.30 Auxiliary current and current of phase 5

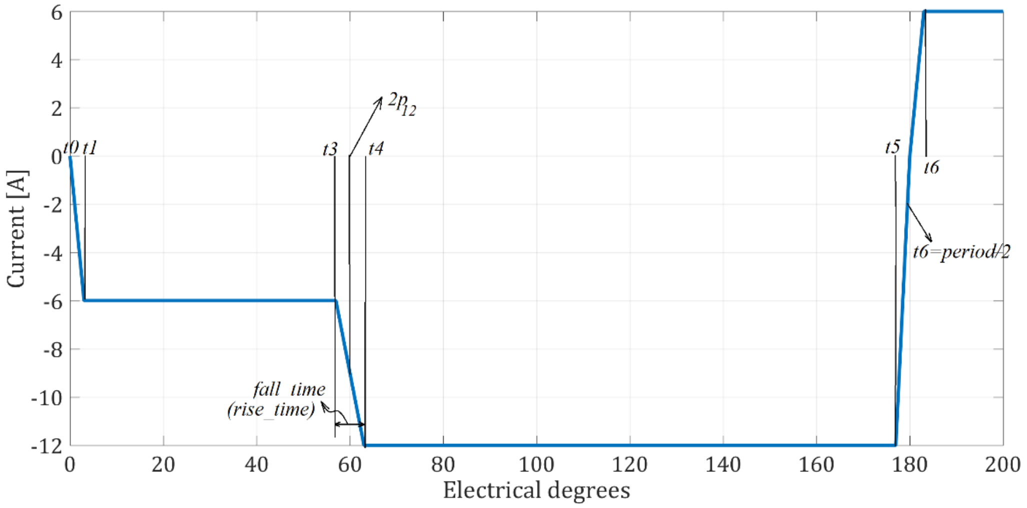


Fig. 2.31 Time instants necessary for definition of auxiliary current

```
t1=0;
v1=0;

t2=rise_time/2;
```

```
v2=-i_field;

t3=2*p12-rise_time/2;
v3=-i_field;

t4=2*p12+rise_time/2;
v4=-i_torque;

t5=period/2-rise_time/2;
v5=-i_torque;

t6=period/2;
v6=0;

t7=period/2+rise_time/2;
v7=i_field;

t8=period/2+2*p12-rise_time/2;
v8=i_field;

t9=period/2+2*p12+rise_time/2;
v9=i_torque;

t10=period-rise_time/2;
v10=i_torque;

t11=period;
v11=0;


t_pwl_help=[t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11];
v_pwl_help=[v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11];

t_pwl_help_6=[t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11];
v_pwl_help_6=[v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11];

t_pwl=[0 t_pwl_help period];
v_pwl=[0 v_pwl_help 0];

%initial shift of phase 5

shift_el_deg=15; %5mech*3
t_pwl_help=t_pwl_help+((shift_el_deg/angular_resolution_el)*time_step)/1000;
t_pwl_help_6=t_pwl_help_6+((shift_el_deg/angular_resolution_el)*time_step)/1000;

for i=1:11

if t_pwl_help_6(i)>period
  t_pwl_help_6(i)=t_pwl_help_6(i)-period;
end

end

for i=1:11

if t_pwl_help(i)<0
  t_pwl_help(i)=t_pwl_help(i)+period;
end

if t_pwl_help(i)>period
  t_pwl_help(i)=t_pwl_help(i)-period;
```

```
end

end

t_pwl=[0 t_pwl_help period];
v_pwl=[i_torque v_pwl_help i_torque];

%current of coil 5

invoke(MN7, 'processCommand', 'REDIM PWLINEAR_1(25)');
%first point
t1_str=num2str(t_pwl(1));
invoke(MN7, 'processCommand', ['PWLINEAR_1(0)=',t1_str]);
v1_str=num2str(v_pwl(1));
invoke(MN7, 'processCommand', ['PWLINEAR_1(1)=',v1_str]);
%second point
t2_str=num2str(t_pwl(2));
invoke(MN7, 'processCommand', ['PWLINEAR_1(2)=',t2_str]);
v2_str=num2str(v_pwl(2));
invoke(MN7, 'processCommand', ['PWLINEAR_1(3)=',v2_str]);
%third point
t3_str=num2str(t_pwl(3));
invoke(MN7, 'processCommand', ['PWLINEAR_1(4)=',t3_str]);
v3_str=num2str(v_pwl(3));
invoke(MN7, 'processCommand', ['PWLINEAR_1(5)=',v3_str]);
%4th point
t4_str=num2str(t_pwl(4));
invoke(MN7, 'processCommand', ['PWLINEAR_1(6)=',t4_str]);
v4_str=num2str(v_pwl(4));
invoke(MN7, 'processCommand', ['PWLINEAR_1(7)=',v4_str]);
%5th point
t5_str=num2str(t_pwl(5));
invoke(MN7, 'processCommand', ['PWLINEAR_1(8)=',t5_str]);
v5_str=num2str(v_pwl(5));
invoke(MN7, 'processCommand', ['PWLINEAR_1(9)=',v5_str]);
%6th point
t6_str=num2str(t_pwl(6));
invoke(MN7, 'processCommand', ['PWLINEAR_1(10)=',t6_str]);
v6_str=num2str(v_pwl(6));
invoke(MN7, 'processCommand', ['PWLINEAR_1(11)=',v6_str]);
%7th point
t7_str=num2str(t_pwl(7));
invoke(MN7, 'processCommand', ['PWLINEAR_1(12)=',t7_str]);
v7_str=num2str(v_pwl(7));
invoke(MN7, 'processCommand', ['PWLINEAR_1(13)=',v7_str]);
%8th point
t8_str=num2str(t_pwl(8));
invoke(MN7, 'processCommand', ['PWLINEAR_1(14)=',t8_str]);
v8_str=num2str(v_pwl(8));
invoke(MN7, 'processCommand', ['PWLINEAR_1(15)=',v8_str]);
%9th point
t9_str=num2str(t_pwl(9));
invoke(MN7, 'processCommand', ['PWLINEAR_1(16)=',t9_str]);
v9_str=num2str(v_pwl(9));
invoke(MN7, 'processCommand', ['PWLINEAR_1(17)=',v9_str]);
%10th point
t10_str=num2str(t_pwl(10));
invoke(MN7, 'processCommand', ['PWLINEAR_1(18)=',t10_str]);
v10_str=num2str(v_pwl(10));
invoke(MN7, 'processCommand', ['PWLINEAR_1(19)=',v10_str]);
%11th point
```

```
t11_str=num2str(t_pwl(11));
invoke(MN7, 'processCommand', ['PWLINEAR_1(20)=',t11_str]);
v11_str=num2str(v_pwl(11));
invoke(MN7, 'processCommand', ['PWLINEAR_1(21)=',v11_str]);
%12th point
t12_str=num2str(t_pwl(12));
invoke(MN7, 'processCommand', ['PWLINEAR_1(22)=',t12_str]);
v12_str=num2str(v_pwl(12));
invoke(MN7, 'processCommand', ['PWLINEAR_1(23)=',v12_str]);
%13th point
t13_str=num2str(t_pwl(13));
invoke(MN7, 'processCommand', ['PWLINEAR_1(24)=',t13_str]);
v13_str=num2str(v_pwl(13));
invoke(MN7, 'processCommand', ['PWLINEAR_1(25)=',v13_str]);
```

Up to now, speed and simulation parameters were just defined in Matlab. First, motion components must be defined and after that simulation parameters are sent to MagNet. The part of code which is communicating those parameters with MagNet is shown below.

```
invoke(MN7, 'processCommand', 'REDIM motion(22)');
       %Components of the motion
invoke(MN7, 'processCommand', 'motion(0)= "Rotor Insulation"');
invoke(MN7, 'processCommand', 'motion(1)= "RotorAirGap"');
invoke(MN7, 'processCommand', 'motion(2)= "Spider"');
invoke(MN7, 'processCommand', 'motion(3)= "Lamination1of pole1"');

invoke(MN7, 'processCommand', 'motion(4)= "Lamination2of pole1"');

invoke(MN7, 'processCommand', 'motion(5)= "Lamination3of pole1"');

invoke(MN7, 'processCommand', 'motion(6)= "Lamination4of pole1"');

invoke(MN7, 'processCommand', 'motion(7)= "Lamination5of pole1"');

invoke(MN7, 'processCommand', 'motion(8)= "Insulation1of pole1"');

invoke(MN7, 'processCommand', 'motion(9)= "Insulation2of pole1"');

invoke(MN7, 'processCommand', 'motion(10)= "Insulation3of pole1"');

invoke(MN7, 'processCommand', 'motion(11)= "Insulation4of pole1"');

invoke(MN7, 'processCommand', 'motion(12)= "Insulation5of pole1"');

invoke(MN7, 'processCommand', 'motion(13)= "Laminationleftup1of pole1"');
invoke(MN7, 'processCommand', 'motion(14)= "Laminationleftup2of pole1"');
invoke(MN7, 'processCommand', 'motion(15)= "Laminationleftup3of pole1"');
invoke(MN7, 'processCommand', 'motion(16)= "Laminationleftup4of pole1"');
invoke(MN7, 'processCommand', 'motion(17)= "Laminationleftup5of pole1"');

invoke(MN7, 'processCommand', 'motion(18)= "Laminationrightup1of pole1"');
invoke(MN7, 'processCommand', 'motion(19)= "Laminationrightup2of pole1"');
invoke(MN7, 'processCommand', 'motion(20)= "Laminationrightup3of pole1"');
invoke(MN7, 'processCommand', 'motion(21)= "Laminationrightup4of pole1"');
invoke(MN7, 'processCommand', 'motion(22)= "Laminationrightup5of pole1"');


invoke(MN7, 'processCommand', 'Call getDocument().makeMotionComponent(motion)');
invoke(MN7, 'processCommand', 'Call getDocument().setMotionSourceType("Motion#1",
       infoVelocityDriven)');
invoke(MN7, 'processCommand', 'CALL getDocument().setMotionType("Motion#1",
       infoRotary)');
invoke(MN7, 'processCommand', 'Call getDocument().setMotionRotaryCenter("Motion#1",
       Array(0, 0, 0))'); %Center  array can be used
invoke(MN7, 'processCommand', 'Call getDocument().setMotionRotaryAxis("Motion#1",
       Array(0, 0, 1))'); %RotatinAxis can be used

invoke(MN7, 'processCommand', 'Call getDocument().setMotionPositionAtStartup("Motion#1",
       0)');
invoke(MN7, 'processCommand', 'Call getDocument().setMotionSpeedAtStartup("Motion#1",
       0)');




invoke(MN7, 'processCommand', 'REDIM Speed(1)');
speed_str=num2str(speed);
invoke(MN7, 'processCommand', ['Speed(0)=',speed_str]);
invoke(MN7, 'processCommand', ['Speed(1)=',speed_str]);

invoke(MN7, 'processCommand', 'REDIM Tim(1)');
time_start_str=num2str(time_start);
time_end_str=num2str(time_end);
time_step_str=num2str(time_step);
invoke(MN7, 'processCommand', ['Tim(0)=',time_start_str]);
invoke(MN7, 'processCommand', ['Tim(1)=',time_end_str]);

invoke(MN7, 'processCommand', 'Call getDocument().setMotionSpeedVsTime("Motion#1", Tim,
       Speed)');
invoke(MN7, 'processCommand', ['Call
       getDocument().setFixedIntervalTimeSteps(',time_start_str,',',time_end_str,','
       time_step_str,')']);
```

2.5. Mesh Generation and Boundaries

As every finite element analysis, the domain of the analysis must be spatially divided into sub-domains i.e. discretized. In 2D simulations sub-domains are triangles whilst in 3D they are tetrahedrons. The meshing of the model also affects the time required for the simulation. Maximum mesh element sizes can be defined for each part of the machine and hence an optimum between time simulation and accuracy can be found. Maximum element sizes are defined in table 2.6. and the mesh in one instant of time is shown in figure 2.32.

Part of the machine	Maximum size of mesh element [mm]
Airgap, air in slot, insulation between laminations, copper	0.1
Lamination	0.3
Stator, Spider	0.5

Table 2.6. Maximum mesh element size for each part of the machine

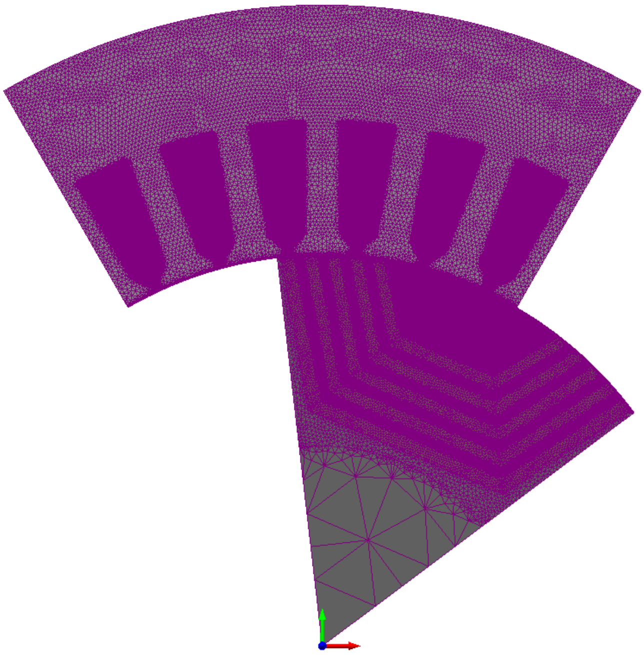


Fig. 2.32 Mesh elements in one of the simulations

As mentioned before, due to the motion, the airgap must be split in two parts. It was also mentioned that it is due to meshing. When motion components are required, like in this case, a so called *re-mesh region* exists. This is the part of the model which is in touch with motion components and it is re-meshed at every time instant due to motion. In our case it is the air gap. According to [12] this type of motion is type C, which means that re-mesh region, i.e. the air gap, must be divided into two parts. One part which will remain static and other one which will move. The rotor air gap is part of the rotor and it rotates together. This makes the process of re-meshing more efficient.

In order to finish the model of the six-phase BLDC reluctance machine periodic boundary conditions must be assigned since only 1/6 of the machine, i.e. one pole, is modeled. In this case odd boundary conditions are necessary. Odd boundary conditions are assigned to the red lines (Fig. 2.33).

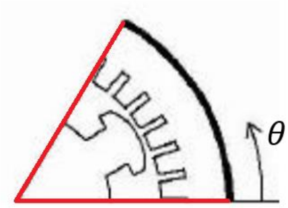


Fig. 2.33 Odd periodicity

$$A_z(r,\theta) = -A_z\left(r,\theta + (2k-1)\frac{\pi}{p}\right), k = 1,2,3,\dots \qquad (2.5)$$

Analytically odd boundary conditions are described with equation (2.5) where A_z is z component of vector magnetic potential. Practically equation (2.5) states that disregarded structures become mirrored images [17]. With adequate meshing and analysis of 1/6 of the machine required time for computation shouldn’t be long.

```
%MESH

max_element_lamination=0.3;
max_element_air_gap=0.1;
max_element_air_slot=0.1;
max_element_insulation=0.1;
max_element_stator=0.5;
max_element_spider=0.5;
max_element_copper=0.1;

invoke(MN7, 'processCommand',['Call
       getDocument().setMaxElementSize("StatorAirGap",',num2str(max_element_air_gap),'
       )']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("RotorAirGap",'
       ,num2str(max_element_air_gap),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Lamination1of
       pole1", ',num2str(max_element_lamination),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Lamination2of
       pole1", ',num2str(max_element_lamination),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Lamination3of
       pole1", ',num2str(max_element_lamination),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Lamination4of
       pole1", ',num2str(max_element_lamination),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Lamination5of
       pole1", ',num2str(max_element_lamination),')']);

invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Laminationleftup1of
       pole1", ',num2str(max_element_lamination),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Laminationleftup2of
       pole1", ',num2str(max_element_lamination),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Laminationleftup3of
       pole1", ',num2str(max_element_lamination),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Laminationleftup4of
       pole1", ',num2str(max_element_lamination),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Laminationleftup5of
       pole1", ',num2str(max_element_lamination),')']);

for i=1:5

   invoke(MN7, 'processCommand',['Call
       getDocument().setMaxElementSize("Laminationrightup',num2str(i),'of pole1",
       ',num2str(max_element_lamination),')']);

end



invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Insulation1of
       pole1", ',num2str(max_element_insulation),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Insulation2of
       pole1", ',num2str(max_element_insulation),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Insulation3of
       pole1", ',num2str(max_element_insulation),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Insulation4of
       pole1", ',num2str(max_element_insulation),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Insulation5of
       pole1", ',num2str(max_element_insulation),')']);

invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("AirSlot1",
       ',num2str(max_element_air_slot),')']);
```

```
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("AirSlot2",
       ',num2str(max_element_air_slot),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("AirSlot3",
       ',num2str(max_element_air_slot),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("AirSlot4",
       ',num2str(max_element_air_slot),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("AirSlot5",
       ',num2str(max_element_air_slot),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("AirSlot6",
       ',num2str(max_element_air_slot),')']);

invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Stator",
       ',num2str(max_element_stator),')']);
invoke(MN7, 'processCommand',['Call getDocument().setMaxElementSize("Spider",
       ',num2str(max_element_spider),')']);
for i=1:P
    invoke(MN7, 'processCommand',['Call
       getDocument().setMaxElementSize("Copper',num2str(i),'",
       ',num2str(max_element_air_slot),')']);
end

%Cancel all contruction lines
invoke(MN7, 'processCommand', 'LinesAndArcs = Array(infoSliceArc, infoSliceLine)');
invoke(MN7, 'processCommand', 'CALL getDocument.getView.selectAll(infoSetSelection,
       LinesAndArcs)');
invoke(view, 'deleteSelection');

%BOUNDARY CONDITIONS

invoke(MN7, 'processCommand', 'REDIM bordi(4)');
invoke(MN7, 'processCommand', 'bordi(0)= "Stator,Face#3"');
invoke(MN7, 'processCommand', 'bordi(1)= "Spider,Face#3"');
invoke(MN7, 'processCommand', 'bordi(2)= "RotorAirGap,Face#3"');
invoke(MN7, 'processCommand', 'bordi(3)= "Rotor Insulation,Face#3"');
invoke(MN7, 'processCommand', 'bordi(4)= "StatorAirGap,Face#3"');
invoke(MN7, 'processCommand', 'Call getDocument.createBoundaryCondition(bordi, "BC1")');

invoke(MN7, 'processCommand', 'REDIM RotationAxis(2)');
invoke(MN7, 'processCommand', 'RotationAxis(0)= 0');
invoke(MN7, 'processCommand', 'RotationAxis(1)= 0');
invoke(MN7, 'processCommand', 'RotationAxis(2)= 1');

invoke(MN7, 'processCommand', 'REDIM Center(2)');
invoke(MN7, 'processCommand', 'Center(0)= 0');
invoke(MN7, 'processCommand', 'Center(1)= 0');
invoke(MN7, 'processCommand', 'Center(2)= 0');
invoke(MN7, 'processCommand','Call getDocument().setOddPeriodic("BC1", Null, -60,
       RotationAxis, Null, Null, Center)');

clear MN7;
```

Chapter 3
Simulations and results

In this chapter, the most significant FEA results of the machine are presented. In the first part of the chapter, the machine is investigated in rated conditions and with different values of field and torque currents. For two simulations $(I_F, I_T) = (12{,}12)A$ and $(I_F, I_T) = (6{,}6)A$ power and efficiency are calculated, flux linkages and voltages are extracted, while for others (section 3.1.4) only average torque and torque ripple are investigated. Furthermore, in section 3.1, the effect of armature reaction is studied. In section 3.2 the main parameters of the machine are found such as resistances and apparent and incremental inductances. It is explained in detail how the incremental inductances can be calculated and the limitation of the presented model. Then torque calculated using FEM and classical analytical formula are compared. In section 3.3 the effect of the air gap length is investigated on the performances of the machine, while in section 3.4 the speed dependent power losses are calculated for different speeds.

Since the same solver has been used in all simulations, it is convenient to explain the choice of solver and motivation behind it. In MagNet 2D simulations there are four types of solvers: static, time harmonic, transient and transient with motion. *Transient with motion* solver has been chosen for analysis of the machine. The benefits of using this solver are numerous. First of all, motion solver reports in global results window set of information related to the motion, but in our case the most important one is torque. Since the currents have trapezoidal shape, transient solver must be used because it allows coils to be driven by any current waveform. Besides torque in the global results window, flux linkages, voltages and energy are reported. Moreover, transient with motion solver accounts all losses in the machine which are related to the motion. Although motion problems can be solved using static solver and parametrisation, the transient with motion solver is more precise. On the other hand, this solver requires stronger computational effort, and on average, 2h were needed to solve a full electric cycle; the size of model files containing the solutions were around 3.5GB each.

3.1 Analysis of the Machine with Various Field (I_F) and Torque (I_T) Currents

In this section, results of simulations with various levels of torque and field currents are explained. All simulations have been performed at rated speed of the machine, i.e. 250rpm. Moreover, all models were solved as transient with motion 2D. This solver is used to capture all transients as well as to calculate the iron losses. Furthermore, the effect of the armature reaction is observed by comparing flux linkages and fields waveforms in load and no-load conditions. Finally, a table with average torque and torque ripple is presented for 36 combinations of I_F and I_T and it is demonstrated how the skewing of rotor can reduce torque ripple.

3.1.1 Field current 12A, torque current 12A

Firstly, the machine will be investigated in rated conditions, i.e. $(I_F, I_T) = (12,12)A$ and 250 rpm (26.18rad/s). The current waveforms are shown in Fig. 3.1, while torque, flux linkages and voltage of phase 1 are presented in Fig. 3.2 – Fig. 3.4. From Fig. 3.2 it can be seen that fluxes are shifted for 30° electrical, which means that the voltages are also shifted by the same amount. However, in Fig. 3.4 only one phase voltage is depicted due to the clarity. This simulation was carried out for a full electric cycle, corresponding to an 80ms time interval.

As mentioned before, torque, voltages and flux linkages are reported after every simulation in global results window. Only the instantaneous electrical power was obtained in post-processing by multiplying related currents and voltages and then summing them up. This waveform is shown in Fig 3.5. It is worth mentioning that all the results in Fig. 3.2 to 3.5 are for one 1/6 of the machine. In the table 3.1. results for full machine are given.

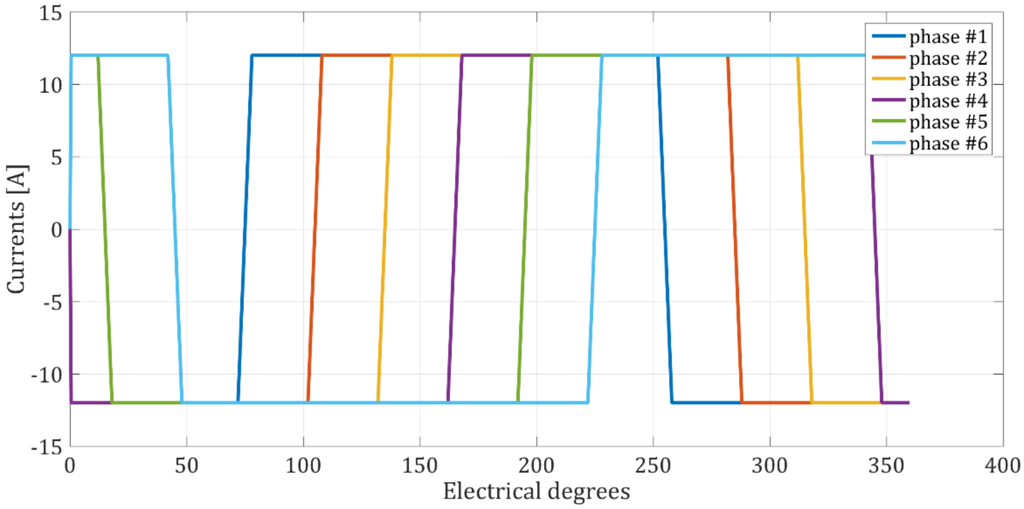


Fig. 3.1 Currents with $(I_F, I_T) = (12,12)A$

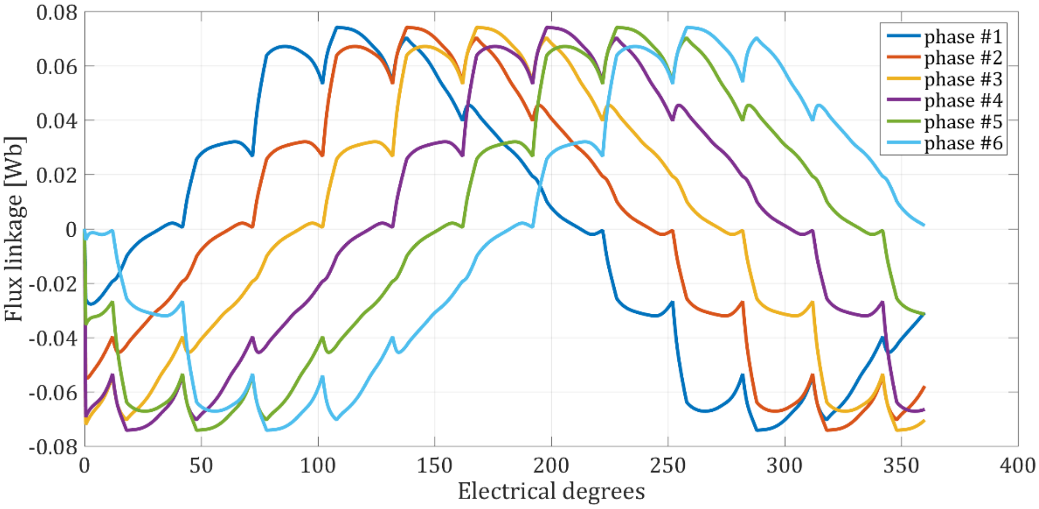


Fig. 3.2 Flux linkages of phase 1 to phase 6

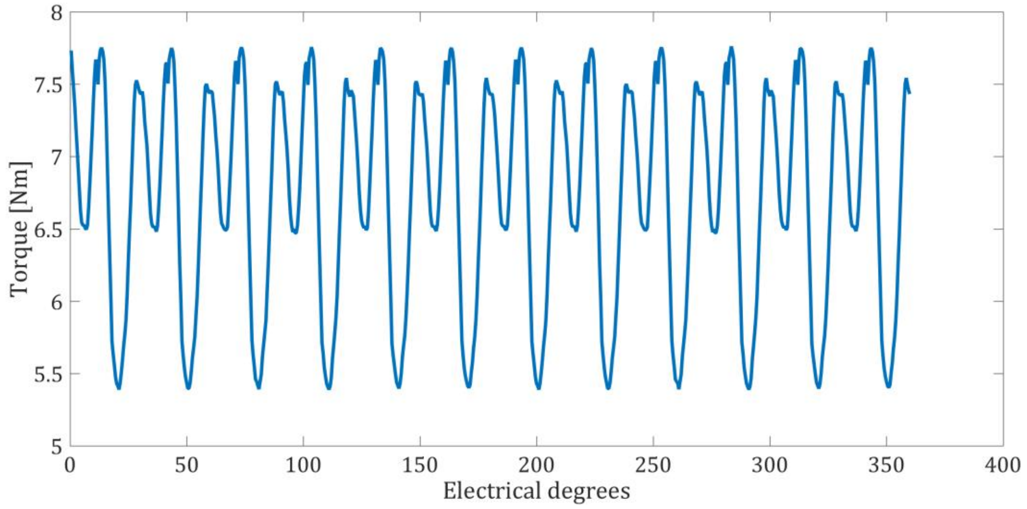


Fig. 3.3 Torque waveform for $(I_F, I_T) = (12,12)A$

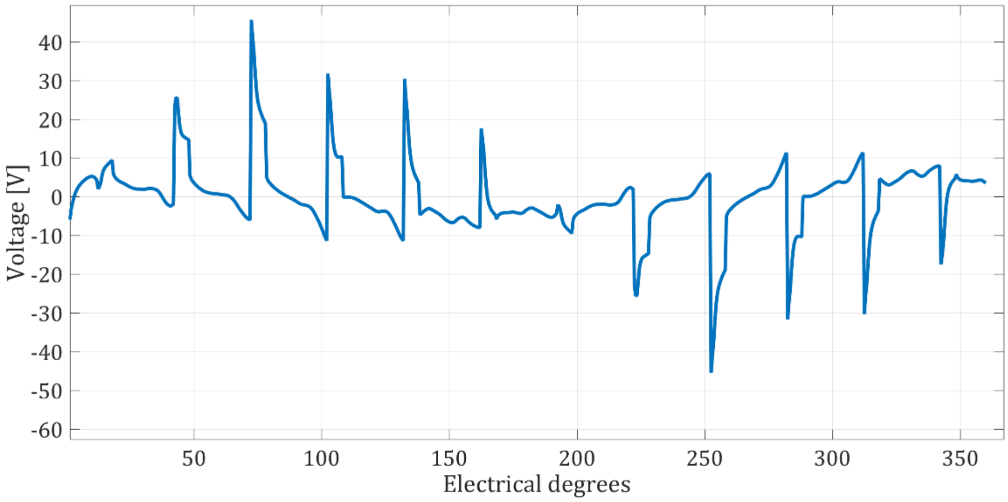


Fig. 3.4 Voltage waveform of phase 1

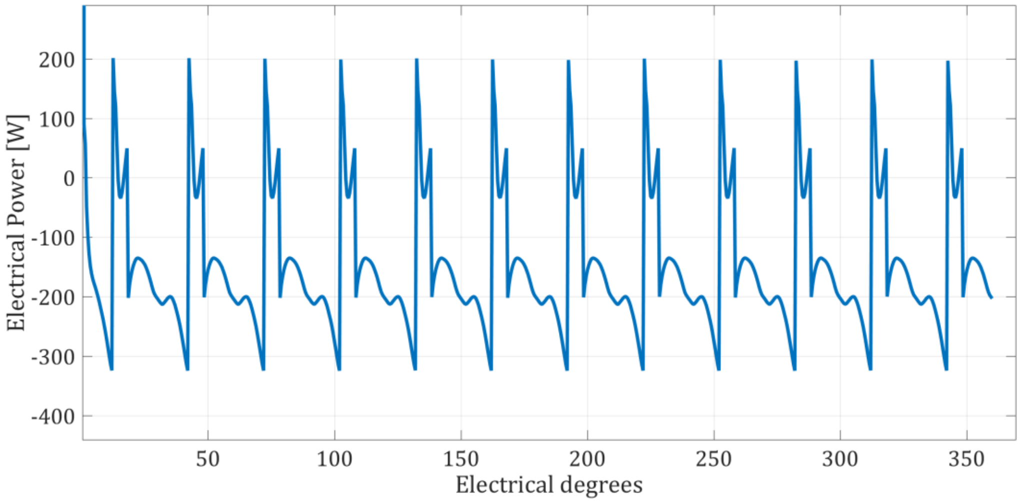


Fig. 3.5 Instantaneous electrical power

	1/6 of the machine	Full machine
Max. flux linkage	74.17 mWb	445.0145 mWb

Average torque (T_{avg})	6.74Nm	40.44 Nm
Torque ripple	2.367 Nm	14.204 Nm
Torque ripple in %	35.17%	35.17%
$P_{mech} = T_{avg}\omega$	176.22 W	1057.3W
P_{el}	148.92 W	893.52 W

Table 3.1 Some machine's results $(I_F, I_T) = (12{,}12)A$

From Fig. 3.5 it can be observed that machine operates in generator mode, i.e. the average electrical power is -150W. Generation mode is actually expected since the electromagnetic torque acts in positive direction of z-axis as reported by MagNet, while speed vector is in the negative direction of z-axis. As it is known that rotational mechanical power can be calculated as $P_{mech} = \vec{T} * \vec{\omega}$ where * is the scalar product, it is obvious that for two vectors in opposite directions power will be negative. Therefore, mechanical power is input to the system while electrical power is output of the system.

Since only one sixth of the machine is simulated the total electrical (output) power is -893.52W (893.52W). As the average torque is 40.44 Nm that means that mechanical i.e. input power is 1057.3W. Since the mechanical and electrical power are known, efficiency and total losses can be calculated. Total losses in the machine are equal to the difference $P_{mech} - P_{el}$ and they account 163.78W, which is equivalent to the efficiency of 84.5% ($\frac{P_{el}}{P_{mech}}100\%$).

Besides torque, flux linkages and voltages which are calculated during solving, transient with motion solver, also calculates ohmic and iron losses as well. Total losses of the machine calculated by the solver are presented in table 3.2. Iron losses are reported for each component for which the loss curve in W/kg is provided and for which the mass density is defined. If some component doesn't comply with those requirements it will be listed with zero iron losses [16]. Total losses according to table 3.2 are 151W which is close to 163.78W previously calculated. This small difference of 12.78W (1.2% of input power) might arise from the numerical calculation of powers in post-processing.

	1/6 of the machine	Full machine
Copper losses phase #1	3.15W	
Copper losses phase #2	3.15W	
Copper losses phase #3	3.15W	
Copper losses phase #4	3.15W	
Copper losses phase #5	3.15W	
Copper losses phase #6	3.15W	
Total (Σ)	18.9W	113.4W
Omhic losses Insulation #1	1.2W	
Omhic losses Insulation #2	1.090W	
Omhic losses Insulation #3	0.516W	
Omhic losses Insulation #4	0.180W	
Omhic losses Insulation #5	0.180W	
Total (Σ)	3.166	19W
Hysteresis stator	2.43W	
Eddy currents stator	0.673W	
Total (Σ)	3.103W	18.618W

Total losses full machine		151.018W

Table 3.2 All losses in the machine with $(I_F, I_T) = (12,12)A$ and 250rpm

3.1.2 Field current 6A, torque current 6A

The following simulation has been done with lower field current of 6A. Also, the torque current is lower $(I_F, I_T) = (6,6)A$. Results are presented in Fig. 3.6-3.9 and in table 3.3. As expected the average torque and the maximum flux linkage are lower.

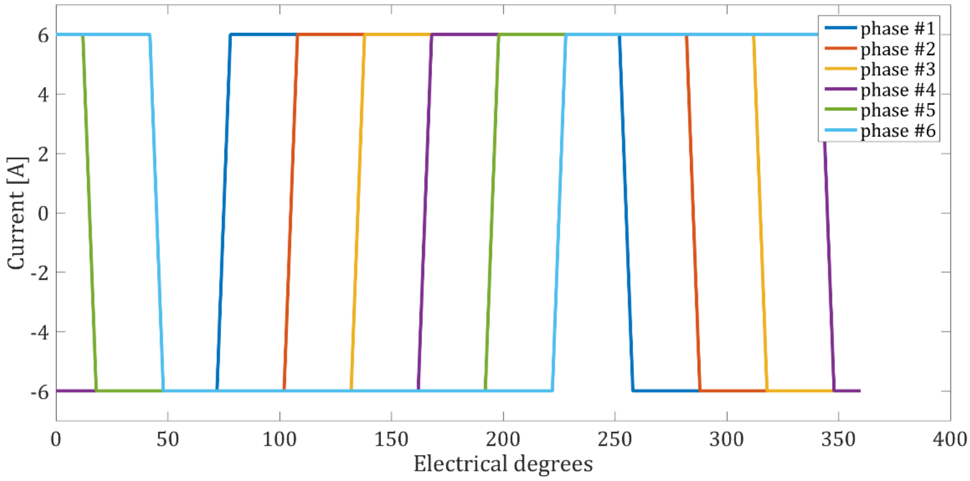


Fig. 3.6 Currents with $(I_F, I_T) = (12,6)A$

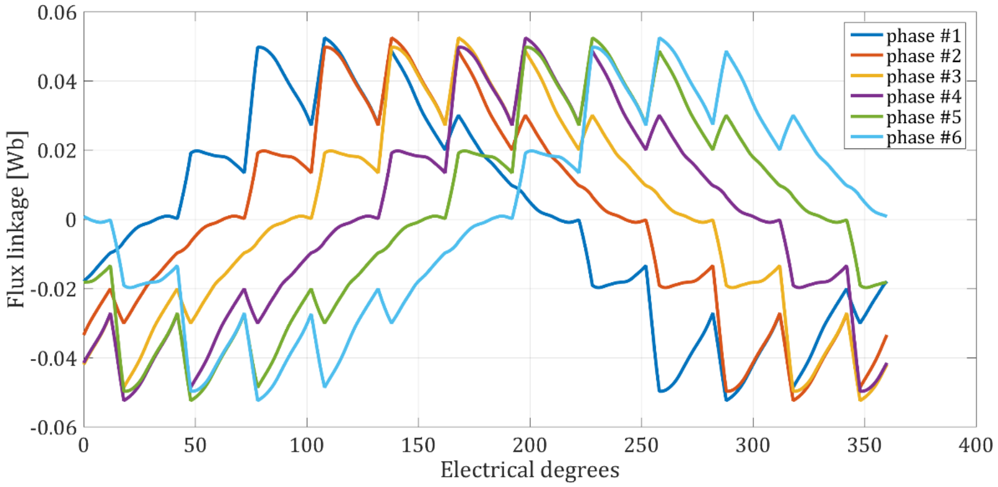


Fig. 3.7 Flux linkages of phase 1 to phase 6

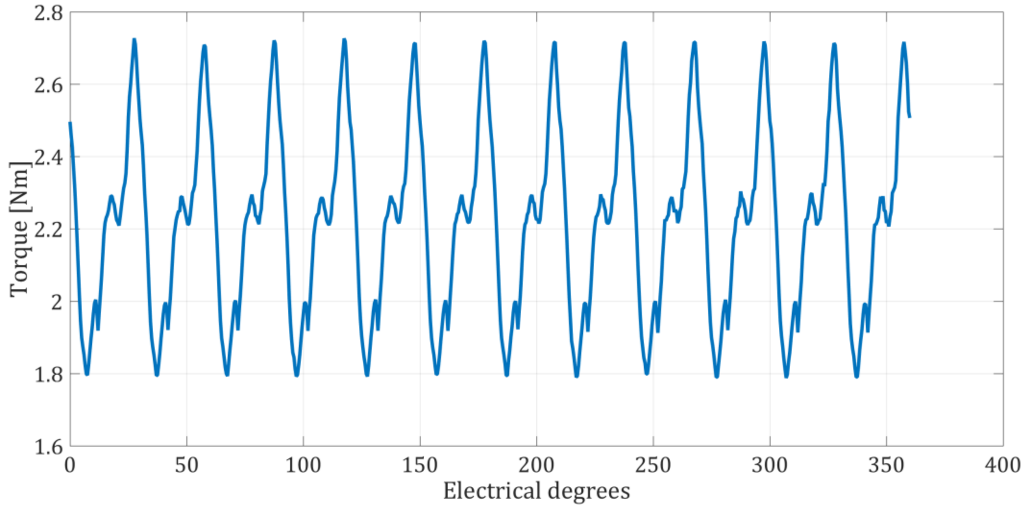


Fig. 3.8 Torque waveform for $(I_F, I_T) = (12,6)A$

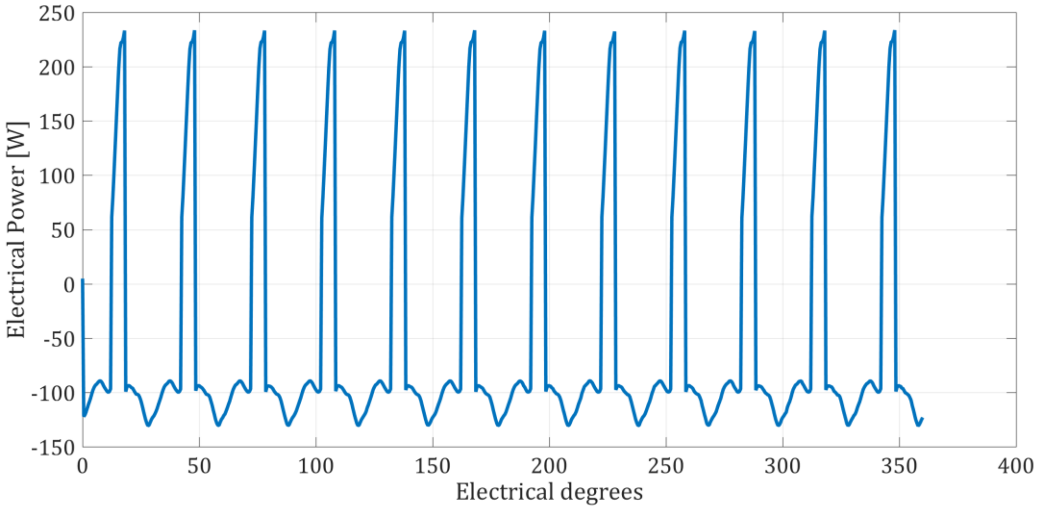


Fig. 3.9 Instantaneous electrical power

	1/6 of the machine	Full machine
Max. flux linkage	52.45 mWb	314.7 mWb
Avg. torque	2.214 Nm	13.283 Nm
Torque ripple	0.935 Nm	5.612 Nm
Torque ripple in %	32.07%	32.07%
$P_{mech} = T_{avg}\omega$	57.95 W	347.75 W
P_{el}	50.13 W	300.78 W

Table 3.3 Some machine's results $(I_F, I_T) = (12,6)A$

As for the previous simulation, total losses and efficiency can be found from table 3.3. Total losses are equal to the difference between input (mechanical) and output (electrical) power, and for this operating point they account for 47.75W, which gives the efficiency of 86.45%. Total losses of the machine, calculated in each component are presented in table 3.4. Once again, these results are available due to the selection of transient with motion solver. Again, good agreement between post-processing and MagNet calculation is obtained. The difference is around 5W (1.4% of input power).

	1/6 of the machine	Full machine
Copper losses phase #1	0.787W	
Copper losses phase #2	0.787W	
Copper losses phase #3	0.787W	
Copper losses phase #4	0.787W	
Copper losses phase #5	0.787W	
Copper losses phase #6	0.787W	
Total (Σ)	4.722W	28.332W
Omhic losses Insulation #1	0.154W	
Omhic losses Insulation #2	0.0507W	
Omhic losses Insulation #3	0.0304W	
Omhic losses Insulation #4	0.0498W	
Omhic losses Insulation #5	0.0229W	
Total (Σ)	0.3078W	1.85W
Hysteresis stator	1.57W	
Eddy currents stator	0.41WW	
Total (Σ)	1.98W	11.88W
Total losses full machine		42.062W

Table 3.4 All losses in the machine with $(I_F, I_T) = (12{,}6)A$ and 250rpm

3.1.3 Armature Reaction

The following two simulations have been carried out with same field currents but with no-load i.e. with 0A torque current. Idea behind these two simulations is to show how load currents affect no-load flux linkage and flux density. This effect is called armature reaction effect like in DC machine.

3.1.3.1 Field current 12A, torque current 0A

In Fig. 3.10 to 3.16 results of this simulation are presented. From Fig. 3.13 average torque is calculated and it equals to 0Nm which is expected since $I_T = 0$. Moreover, this confirms that the waveforms of the currents have been defined correctly, i.e. all commutation finish and start at the same time. If this isn't the case, circled region in Fig. 3.11, there is a small period of time where both phases carry field, when it is obvious that one must carry torque current and other field current. In this case even if the torque current is zero, due to irregular commutation supposed field currents will for a small portion of time be torque currents, and hence will provide a non-zero average torque.

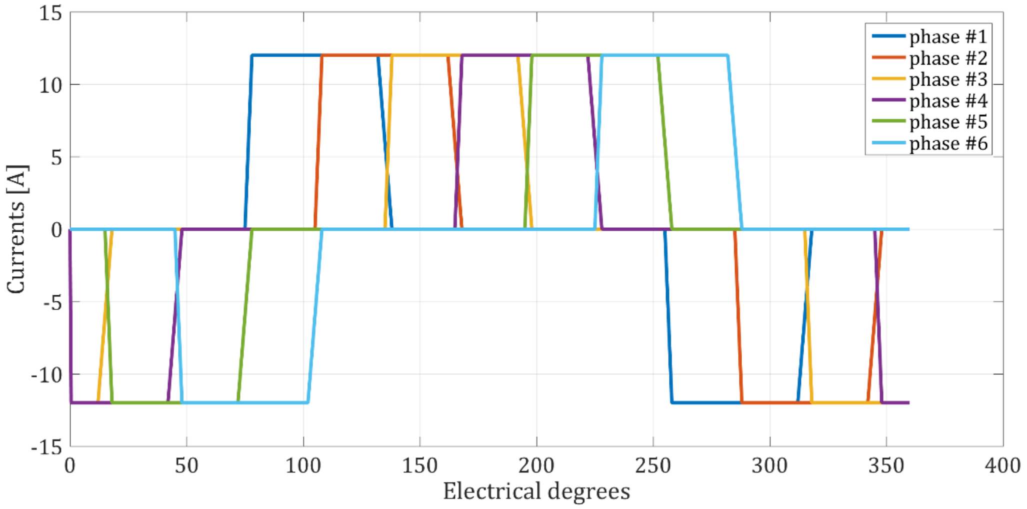


Fig. 3.10 Currents with $(I_F, I_T) = (12{,}0)A$

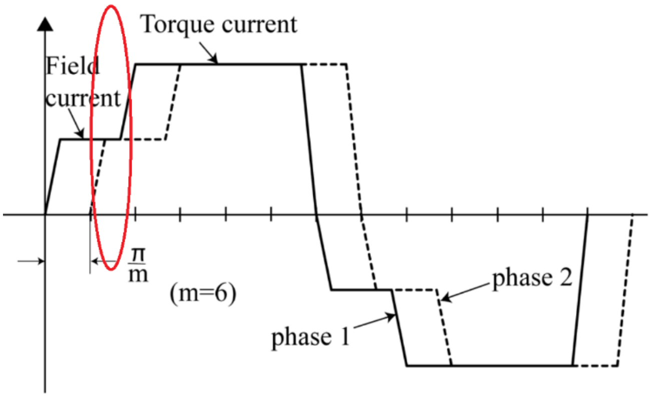


Fig. 3.11 Currents of the machine with irregular commutation

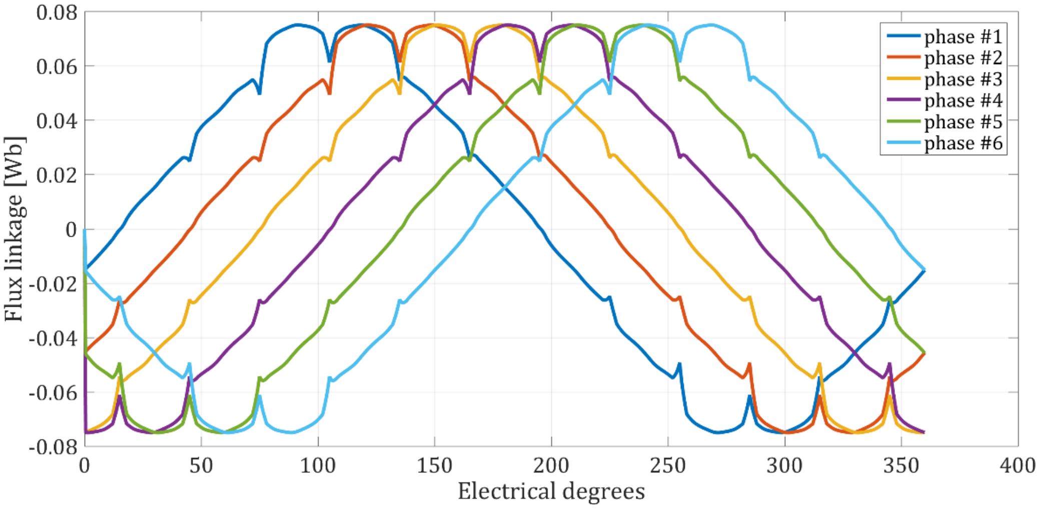


Fig. 3.12 Flux linkages in no-load condition with $I_F = 12A$

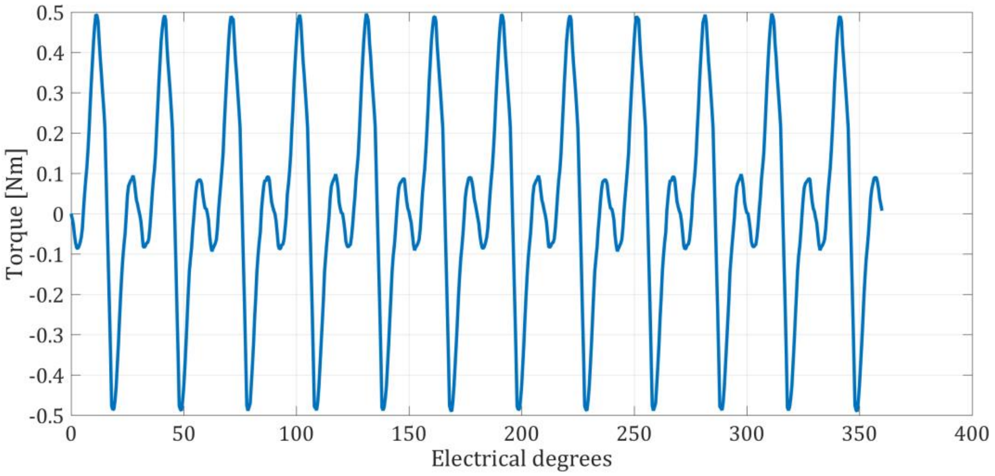


Fig. 3.13 Torque waveform with $(I_F, I_T) = (12{,}0)A$

If figures 3.12 and 3.7 are compared a noticeable difference is observed in the shape of the flux linkage waveforms. This is due to the presence of torque current. This effect can be seen nicely if the flux densities in the middle of the air gap are compared for the load and no-load conditions. Hence, absolute value of normal smoothed *B* field has been measured at 20ms in the middle of the air gap (Fig. 3.14).

In figures 3.15 and 3.16 flux lines can be observed. Without load current flux lines follow the lamination, i.e. they are parallel with it. When load is present, flux lines aren't parallel to lamination, thus due to armature reaction, direction of the field is angled towards *q*-axis.

Another way to see the armature reaction is through flux density. The absolute value of the normal smoothed B field is measured by using Magnet's arc probe tool. In Fig. 3.17 the no-load field is presented and it is almost flat with average trend of 1.05T. However, when the load is present the flux density waveform gets distorted (Fig. 3.18).

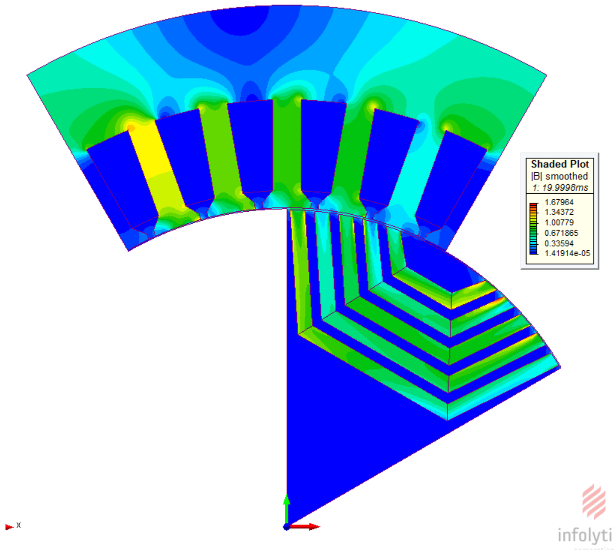


Fig. 3.14 Position of the rotor at 20ms

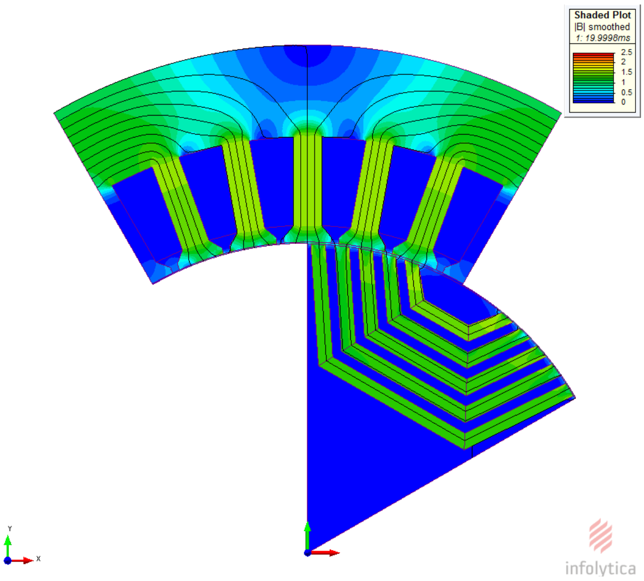


Fig. 3.15 Flux lines at 20 ms with no-load conditions

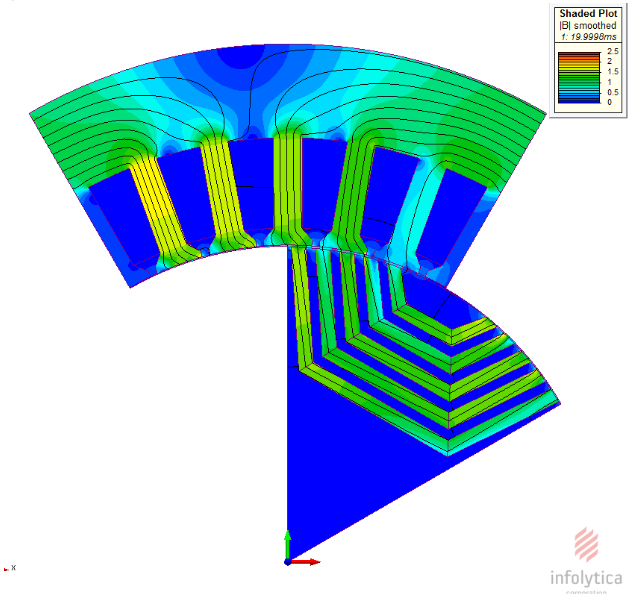


Fig. 3.16 Flux lines under load $I_T = 12A$

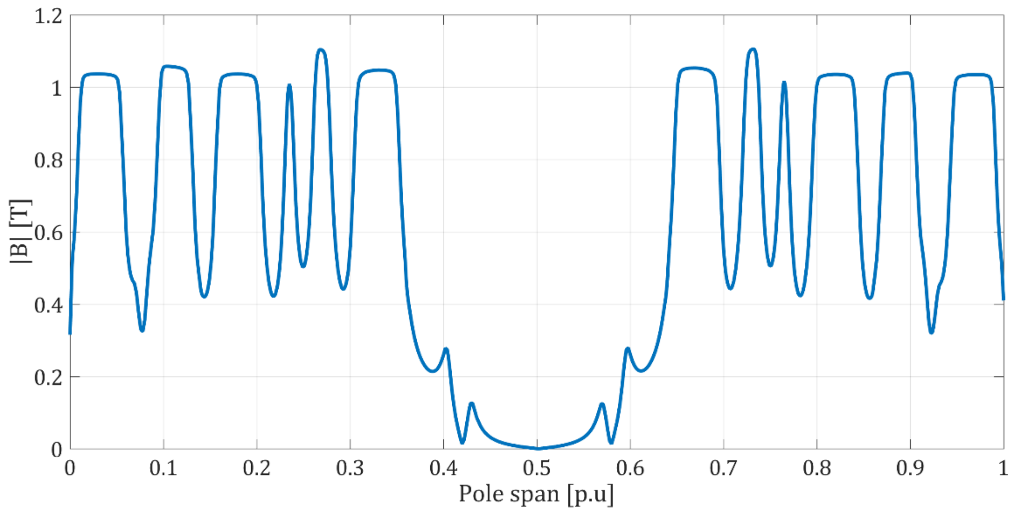


Fig. 3.17 No-load flux density in the middle of the air gap

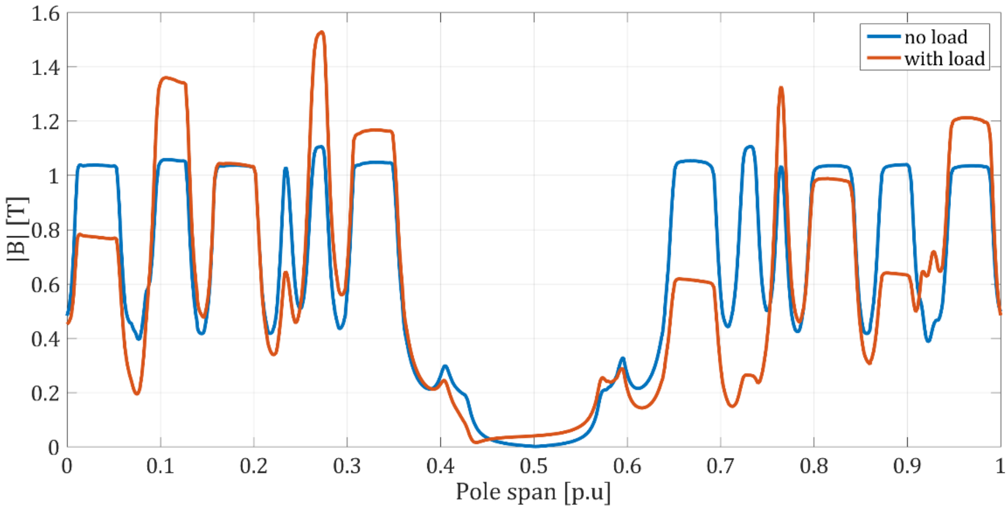


Fig. 3.18 No-load and full-load ($I_T = 12A$) flux densities in the middle of the air gap

3.1.3.2 Field torque 6A, torque current 0A

Results of this simulation are presented in figures from 3.19 to 3.23. Again, it can be noticed how the maximum flux linkage is lower in no-load conditions and how it has a more trapezoidal shape. The average torque is zero which again confirms good definition of current waveforms. Regarding the armature reaction only flux density in the middle of the air gap at 20ms has been measured. No-load flux density is almost flat at around 0.6T (Fig. 3.22). However, when load current is present severe armature reaction is observed. Although torque current is necessary for torque production, it creates the armature reaction which causes some unwanted effects, such as flux density distortion.

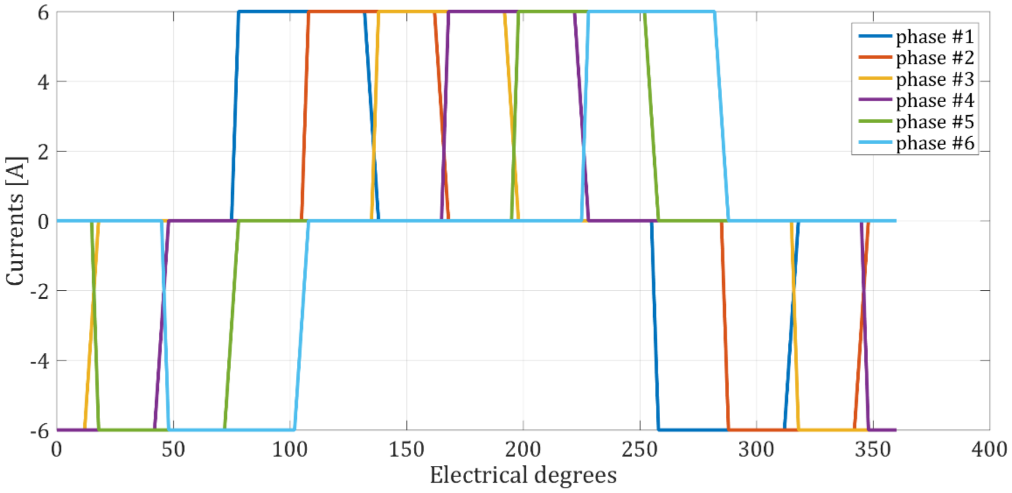


Fig 3.19 Currents with $(I_F, I_T) = (6,0)A$

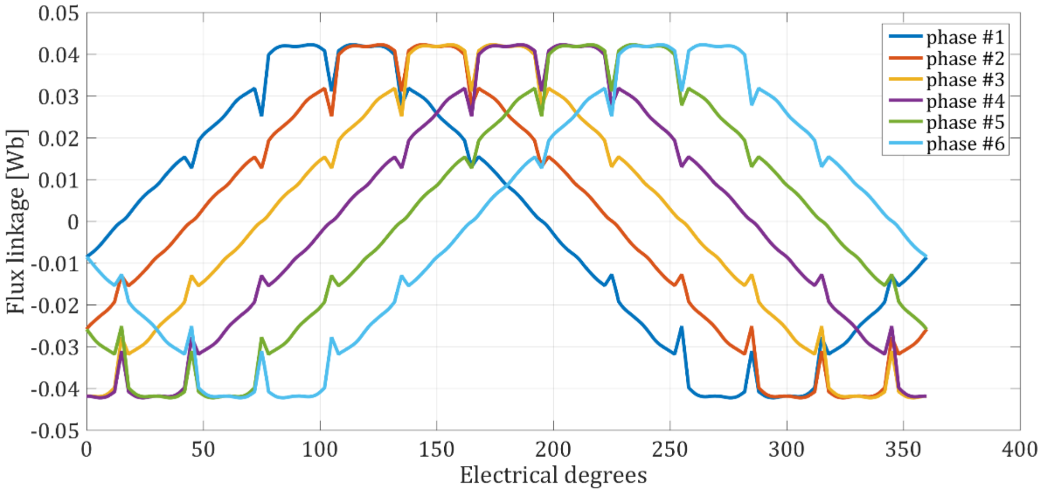


Fig. 3.20 No-load flux densities with $I_F = 6A$

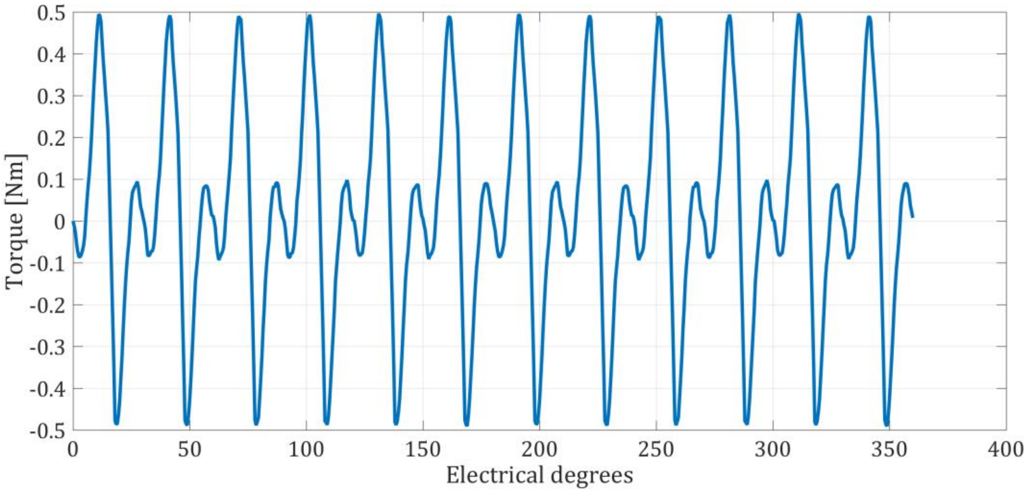


Fig 3.21 Torque waveform with $(I_F, I_T) = (6,0)A$

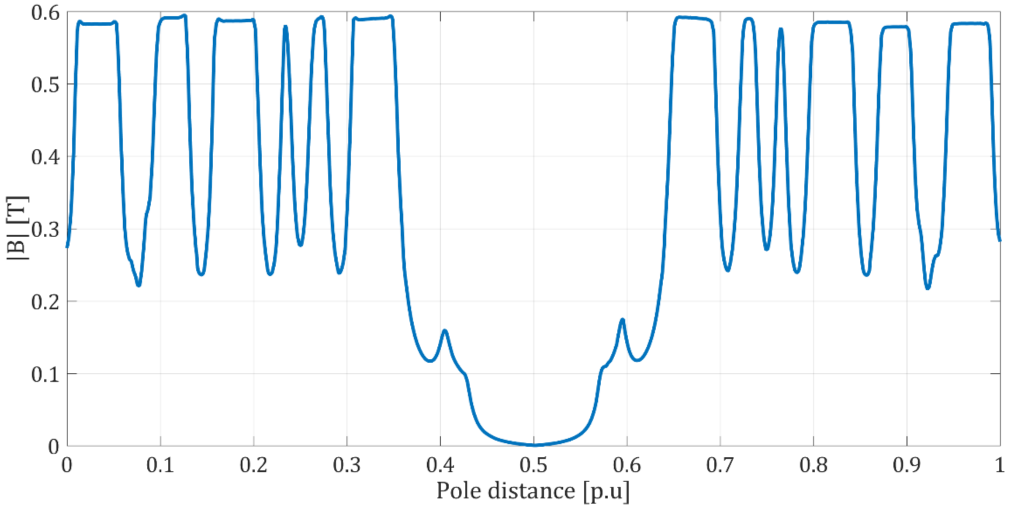


Fig. 3.22 No-load flux density wit $I_F = 6A$

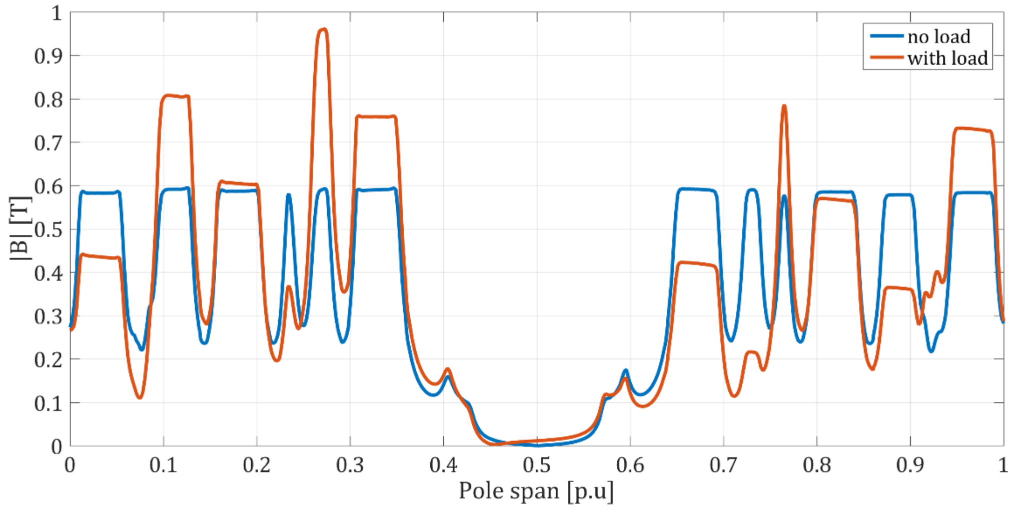


Fig. 3.23 Flux densities with $I_F = 6A$, and load current of 6A and 0A.

3.1.4 Average Torque and Ripple

Besides four presented simulations, another 34 simulations with different field and torque current were carried out. They will not be presented in detail like four simulations explained above. Only average torque and ripple has been investigated. Results are presented in table 3.5 and also in figure 3.24-3.30. Results with zero torque current aren't mentioned. All results are presented for full machine.

I_F $[A]$	$I_T[A]$	T_{avg} $[Nm]$	ΔT $[Nm]$	$\Delta T_{\%}$
12	12	40.385	14.204	35.2
12	10	34.505	17.061	49.4
12	8	28.312	19.868	70.2
12	6	21.659	21.803	100.7
12	4	14.727	22.892	155.4
12	2	7.661	23.389	305.3
10	12	36.706	10.324	146.5
10	10	31.486	10.227	75.1
10	8	25.879	19.868	99.8
10	6	19.909	13.165	50.9
10	4	13.609	22.892	72.7
10	2	7.045	16.391	44.7
8	12	31.601	16.231	272.2
8	10	27.093	11.039	95.2
8	8	22.229	7.920	46.5
8	6	17.049	7.036	31.7
8	4	11.602	7.953	29.4
8	2	5.962	9.863	31.2
6	12	25.228	21.508	477.2
6	10	21.505	14.483	162.3
6	8	17.502	9.313	70.1
6	6	13.283	5.612	32.1

6	4	8.922	3.442	16.0
6	2	4.507	4.985	19.8
4	12	17.788	28.933	967.3
4	10	15.021	20.227	337.9
4	8	12.082	12.702	140.6
4	6	9.035	6.372	52.7
4	4	5.987	2.548	17.0
4	2	2.991	1.689	9.5
2	12	9.662	33.521	2239.1
2	10	8.022	23.845	783.5
2	8	6.319	15.244	327.7
2	6	4.652	8.332	131.8
2	4	3.043	3.364	41.9
2	2	1.497	0.638	6.6

Table 3.5 Average torque and torque ripple for various (I_F, I_T)

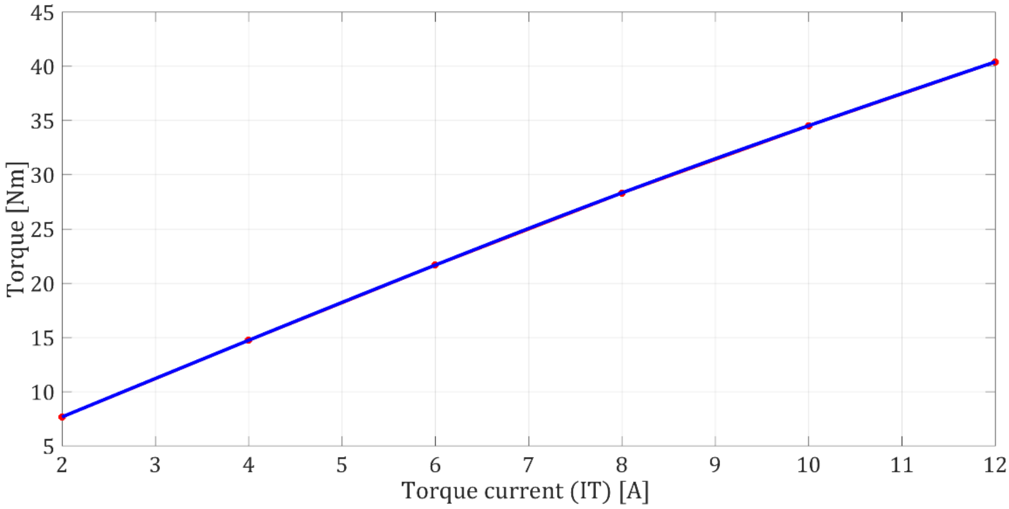


Fig. 3.24 Torque as a function of I_T with $I_F = 12A$

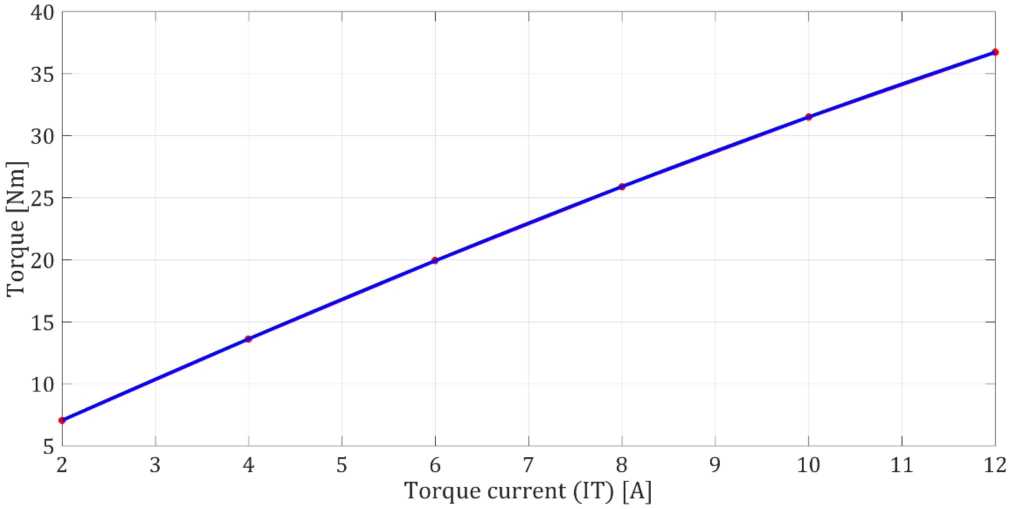


Fig. 3.25 Torque as a function of I_T with $I_F = 10A$

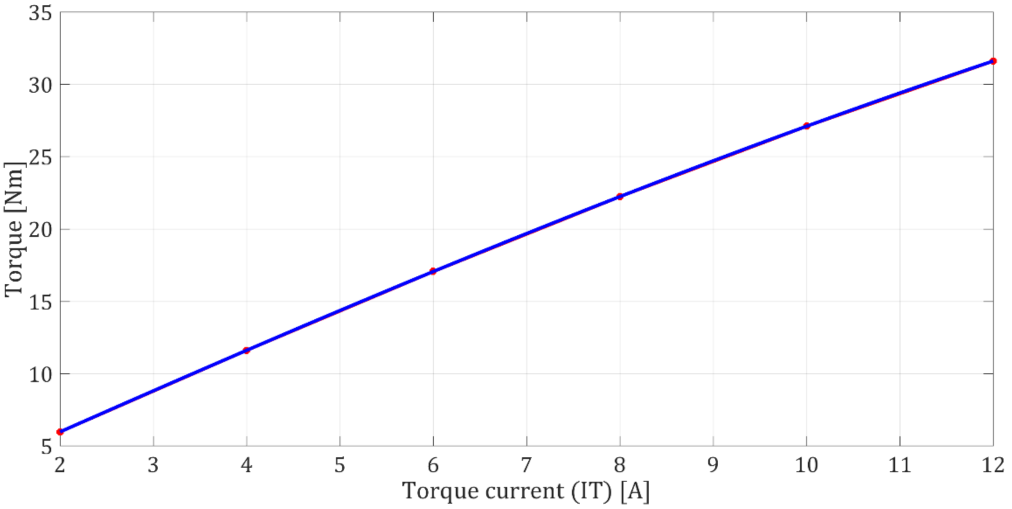


Fig. 3.26 Torque as a function of I_T with $I_F = 8A$

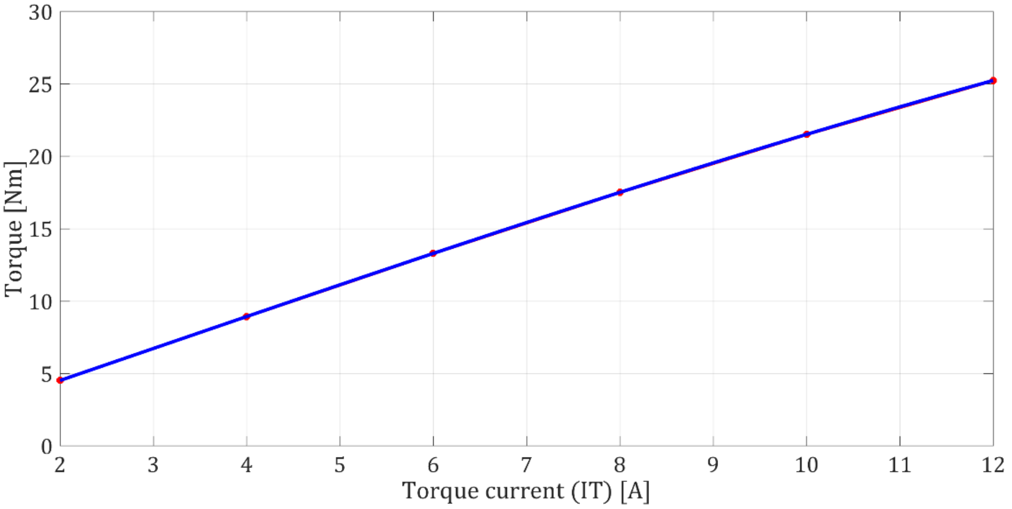


Fig.3.27 Torque as a function of I_T with $I_F = 6A$

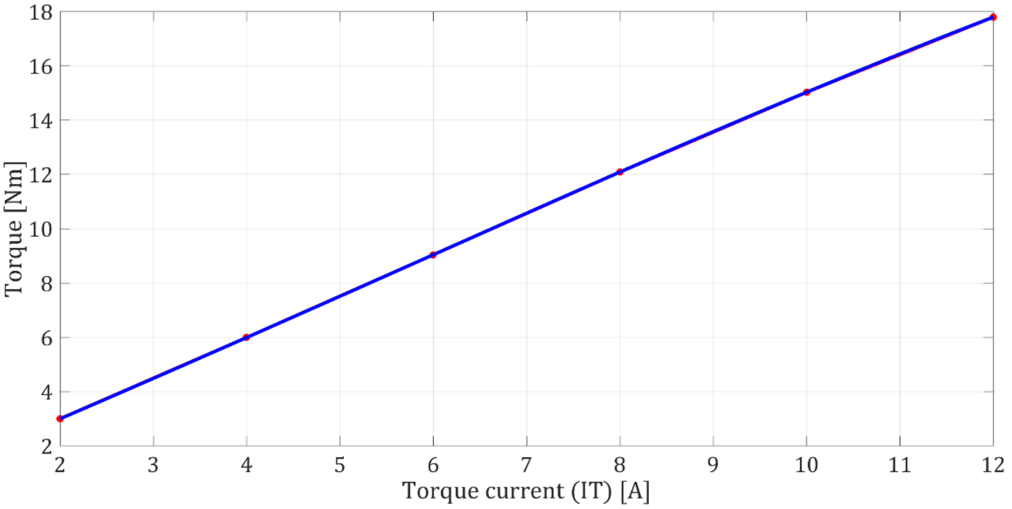


Fig. 3.28 Torque as a function of I_T with$I_F = 4A$

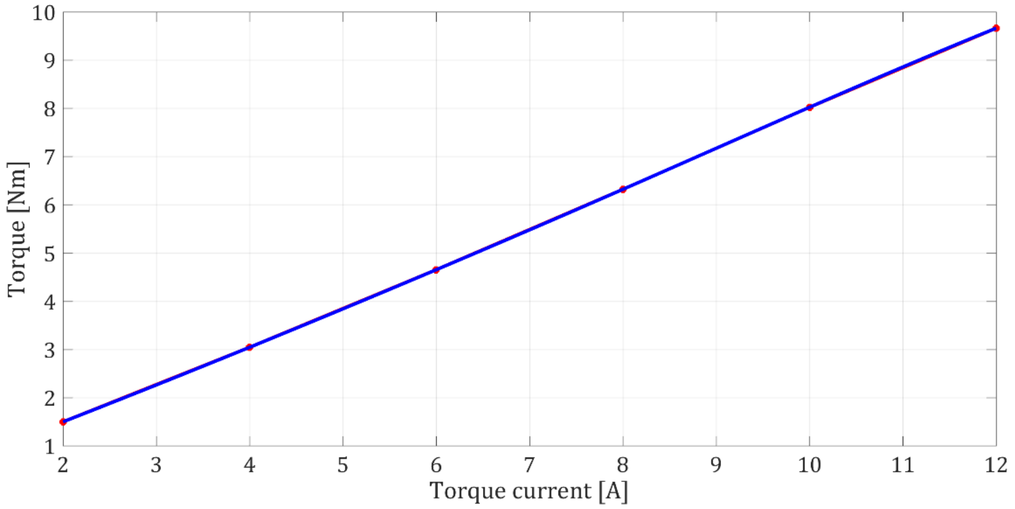


Fig. 3.29 Torque as a function of I_T with$I_F = 2A$

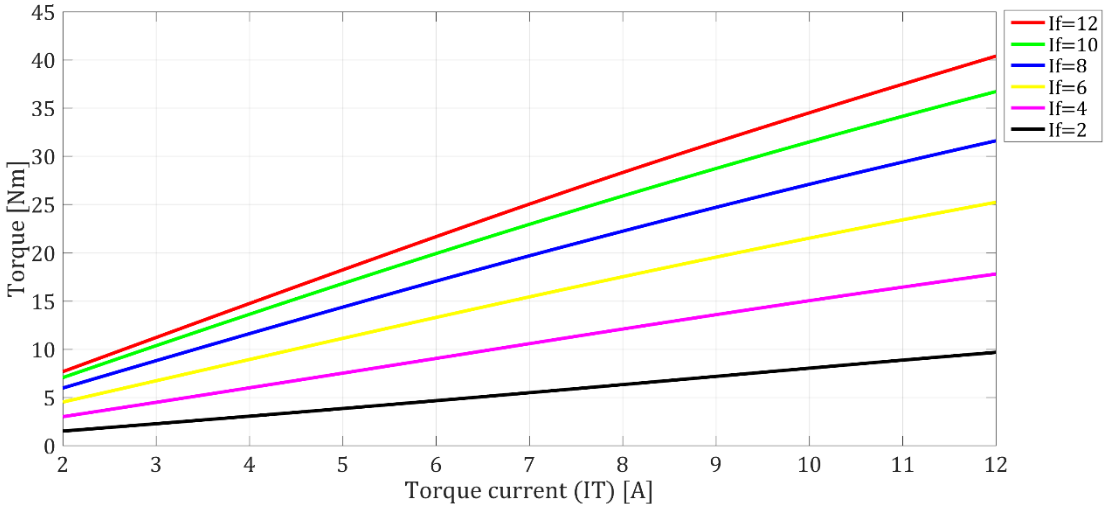


Fig. 3.30 Torque as a function of I_T for various field currents

Torque ripple can be reduced with rotor skewing. For example, let's consider torque with $(I_F, I_T) = (8{,}4)A$. Without skewing, the ripple is 29.35% and the torque waveform is presented in figure 3.31. For instance, let's consider rotor with three segments. Skewed torque is calculated according to formula (3.1)

$$T_{skew} = \frac{1}{k}\sum_{i=1}^{k} T(\theta - (k-1)\frac{\theta_{slot}}{k}) \tag{3.1}$$

where $T(\theta)$ is torque waveform from Fig. 3.31, θ_{slot} angle of the slot, which is in our case 10° mechanical and k is number of rotor segments, in our case 3. When skewing is performed, with 3 rotor segments, we get waveform at Fig.3.32. Reduction in ripple is obvious: from 7.953Nm to 1Nm. Average value with skewed rotor is 11.56Nm which is slightly lower than 11.6Nm without skewing. Ripple in percentage is now 8.65%.

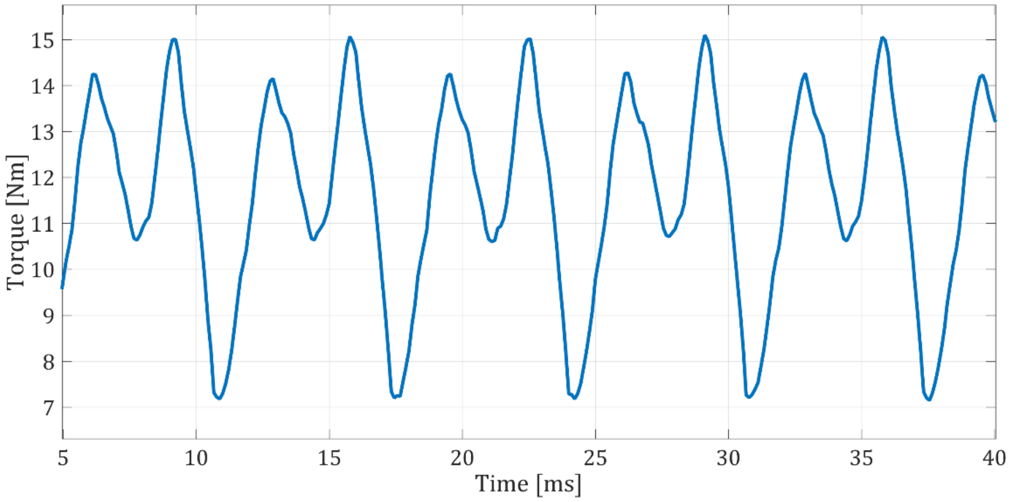


Fig. 3.31 Torque waveform with $(I_F, I_T) = (8{,}4)A$

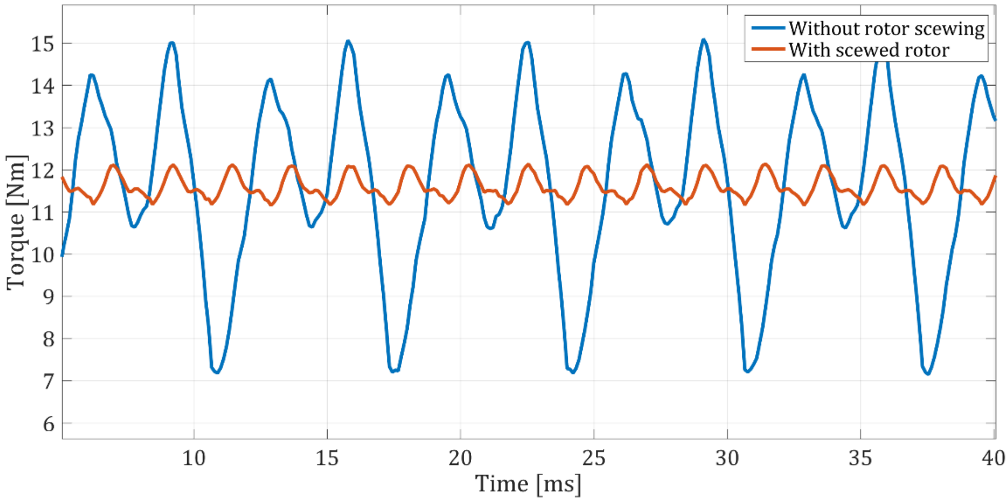


Fig. 3.32 Torque waveform with $(I_F, I_T) = (8{,}4)A$ without and with rotor skewing

3.2. Parameter Identification

In this part of the chapter, parameters of the machine resistance and inductance will be calculated. Simulations for inductances calculation will be explained and results will be presented. Machine's parameters are important for machine modelling and for current control. Also in this part of the chapter it will be shown how and when calculated parameters could be used for finding torque and what are the limits of calculated parameters.

3.2.1 Resistance

Firstly, the resistance was identified experimentally on the available machine. The four wire kelvin method was used. Current of 2A was fed into one phase of the machine and voltage of 0.806V was measured. Thus, measured resistance is 0.403Ω $(\frac{U}{I} = \frac{0.806V}{2A})$. This resistance includes the resistance of the end windings and of the connections.

Another method of phase resistance identification is through copper losses, which were calculated in MagNet. From previously presented simulations, copper losses for current of 12A are 3.2195W. Using RI^2 formula, resistance can be calculated, $R = 0.0223\Omega$ (only one sixth of the resistance)

Phase resistance can be also calculated using well-known formula (3.2)

$$R = \rho \frac{l}{S} \tag{3.2}$$

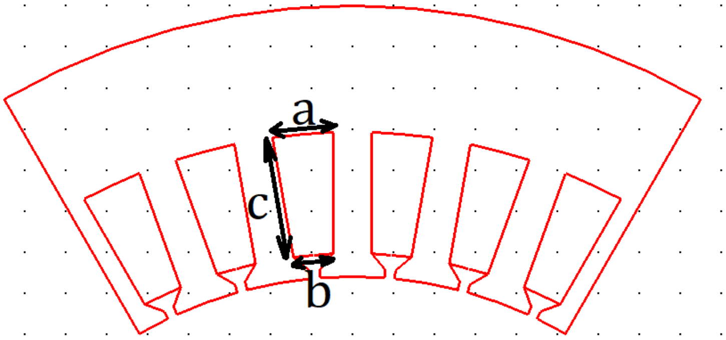


Fig. 3.33 Dimensions of the slot

In our case $l = 150$mm and S can be calculated. Slot area is trapezoid with edges $a = 7.45\ mm, b = 4.86\ mm$ and $c = 14.8\ mm$ (Fig.3.33). Using formula for the area of trapezoid, slot area is computed and it equals to 90.74 mm^2. Considering that 28 conductors are embedded in the slot, the area of one conductor is hence 28 times less which is 3.24 mm^2. In one slot, there are 28 conductors in series therefore,

$$R = 28 \cdot 1.68 \cdot 10^{-8} \cdot \frac{150 \cdot 10^{-3}}{3.24 \cdot 10^{-6}} = 0.02177\Omega$$

which is very similar to resistance calculated using copper losses.

As mentioned only 1/6 of the resistance has been calculated, therefore total resistance is six times bigger, i.e. 0.13 Ω. Obviously, this is very different from measured resistance, but slot fill factor wasn't considered when resistance was calculated using (3.2). If end windings are neglected slot fill factor can be calculated as $\frac{R_{calc}}{R_{meas}}$ and it equals to 0.32 $(\frac{0.13}{0.403})$. However, real slot factor is higher than 0.32 since in 0.403 Ω end windings and connection are comprised.

3.2.2 Inductances

As mentioned before the parameters of the per-phase equivalent circuits are important for current control, torque calculation etc. In this section, inductances will be calculated. The classical definition of the inductance is

$$L_{kj} = \frac{\psi_k}{i_j} \tag{3.3}$$

where ψ_k is the flux linkage of *k-th* coil due to the current i_j in *j-th* coil. If $j \neq k$ it is called *mutual inductance* and if $i = k$ it is called *self-inductance*. Inductances defined in this way are also called *apparent inductances*. This definition practically tells that flux linkage in the coil is linearly proportional to the current. If more coils are present the flux linkage is equal to a linear combination of currents (principle of superposition). This is true if conditions for linearity are fulfilled. However, in electrical machines various non-linear materials are used, hence this definition sometimes doesn't give realistic values. In general, in electrical machines flux linkage is non-linear function of currents and rotor position (3.4) and both currents and rotor position are function of time.

$$\psi = f(i, \theta_r) \tag{3.4}$$

$$v = Ri + \frac{d\psi}{dt} \tag{3.5}$$

The voltage balance in every coil is expressed with equation (3.5). Since ψ is function of currents and rotor position, for its time derivative the chain rule formula must be used [18], and therefore, voltage equation of the coil can be written and expanded as:

$$\frac{d\psi}{dt} = \frac{\partial\psi}{\partial i}\frac{\partial i}{\partial t} + \frac{\partial\psi}{\partial\theta_r}\frac{\partial\theta_r}{\partial t} \tag{3.6}$$

Moreover, i is vector of six currents, hence final formula (for i-th coil) is

$$\frac{d\psi_i}{dt} = \frac{\partial\psi_i}{\partial i_1}\frac{\partial i_1}{\partial t} + \frac{\partial\psi_i}{\partial i_2}\frac{\partial i_2}{\partial t} + \frac{\partial\psi_i}{\partial i_3}\frac{\partial i_3}{\partial t} + \frac{\partial\psi_i}{\partial i_4}\frac{\partial i_4}{\partial t} + \frac{\partial\psi_i}{\partial i_5}\frac{\partial i_5}{\partial t} + \frac{\partial\psi_i}{\partial i_6}\frac{\partial i_6}{\partial t} + \frac{\partial\psi_i}{\partial\theta_r}\omega_r \tag{3.7}$$

where $\frac{\partial\theta_r}{\partial t}$ is written as ω_r. Therefore, equations for six-phase machine are

$$[v] = [R][i] + [L]\frac{\partial}{\partial t}[i] + [e] \tag{3.8}$$

where,

$$[v] = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix}, [R] = diag(R), [i] = \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \end{bmatrix} \tag{3.9}$$

$$[L] = \left[L_{ij}\right]_{6\times 6}, L_{ij} = \frac{\partial\psi_i}{\partial i_j} \tag{3.10}$$

$$[e] = [e_i]_{6\times 1}, e_i = \frac{\partial\psi_i}{\partial\theta_r}\omega_r \tag{3.11}$$

Inductances defined as in (3.10) ($L_{ij} = \frac{\partial\psi_{i_i}}{\partial i_j}$) are called *incremental inductances*. Incremental inductances are more precise and they describe relation between flux and currents more accurately. This can be seen from $\psi - i$ curve shown in Fig. 3.34 [19]. For instance, let's consider flux linkage with current i_0. The apparent inductance is the slope of the line connecting the origin and the point with coordinates (i_0, ψ_0). Using that inductance, calculated flux linkage with current $\frac{i_0}{2}$ would be $\frac{\psi_0}{2}$. Obviously, as it can be seen from Fig. 3.34, there is an error in this approach when the nonlinearity is present. On the other hand, incremental inductance is derivation of flux linkage $(\frac{\partial \psi}{\partial i})$, hence it is slope of tangent line in the point of calculation. Therefore, mathematically speaking, this means that incremental inductance depends on operating point. For comparison, derivatives calculated in points i_0 and $\frac{i_0}{2}$ will not be the same. One important thing to be remembered from this brief reminder on inductances, is that for calculation of the incremental inductances *operating point* of the currents must be known.

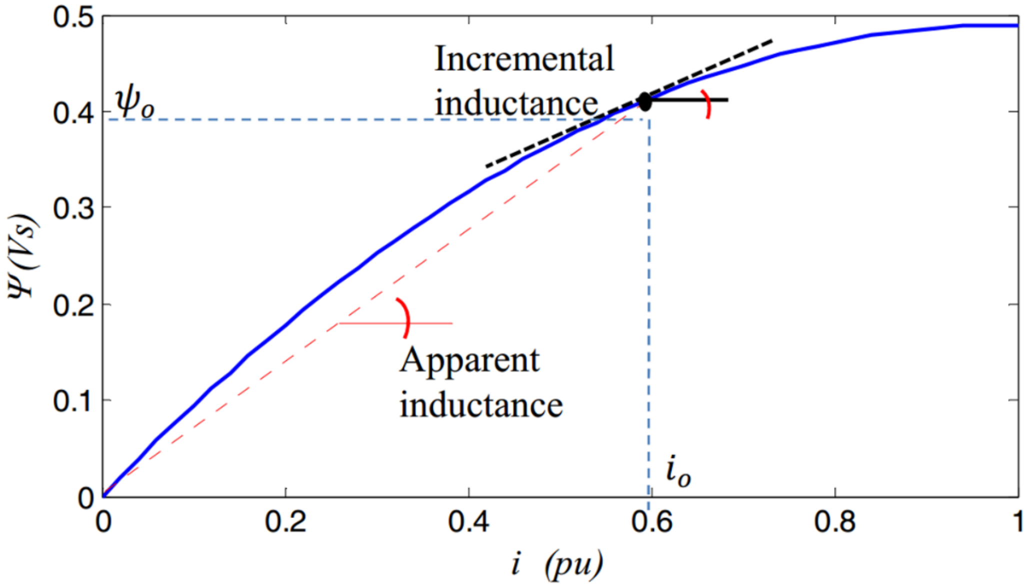


Fig. 3.34 $\psi - i$ curve, apparent and incremental inductances

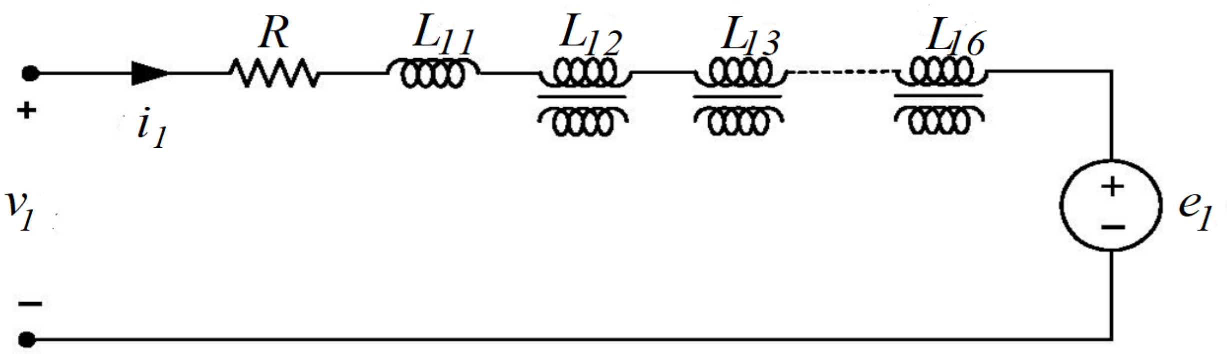


Fig. 3.35 Equivalent circuit of the equation (3.7) for phase 1

Another important remark is that partial derivatives must be calculated. They differ from total derivatives in a way that partial derivatives are calculated by keeping some variables constant and others not. In order to better understand what is kept constant in partial derivatives mathematical notation for this explanation will be used. Let's consider that f is a function of two variables $f(x, y)$ (e.g. f -flux linkage, x -current, y -angle), where x and y are dependent on another variable t (e.g. time) (3.12). Derivative in respect to time is calculated using chain rule (3.13). For convenience notation of partial and time derivatives will be as in (3.14), hence equation (3.13) can be re-written as shown in (3.15).

$$f = f(x,y), x = g(t), y = h(t) \tag{3.12}$$

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{\partial g}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial h}{\partial t} \tag{3.13}$$

$$\frac{\partial f}{\partial x} = f_x(x,y), \frac{\partial f}{\partial y} = f_y(x,y), \frac{\partial g}{\partial t} = g'(t), \frac{\partial h}{\partial t} = h'(t) \tag{3.14}$$

$$\frac{df}{dt} = f_x(x,y)\, g'(t) + f_y(x,y)h'(t) \tag{3.15}$$

To find differential from equation (3.13) in some instant of time t_o, partial derivatives $f_x(x,y)$ and $f_y(x,y)$ must be calculated in point (x_o, y_o) where $x_o = g(t_o), y_o = h(t_o)$.

$$\frac{df}{dt}(t_o) = f_x(x_o,y_o)\, g'(t_o) + f_y(x_o,y_o)h'(t_o) \tag{3.16}$$

Partial derivatives from (3.16) at point (x_o, y_o) can be approximated as in (3.17) and (3.18) [18].

$$f_x(x_o,y_o) = \lim_{h\to 0}\frac{f_x(x_o+h,y_o) - f_x(x_o,y_o)}{h} \approx \frac{f_x(x_o+h,y_o) - f_x(x_o,y_o)}{h} \tag{3.17}$$

$$f_y(x_o,y_o) = \lim_{h\to 0}\frac{f_x(x_o,y_o+h) - f_x(x_o,y_o)}{h} \approx \frac{f_x(x_o,y_o) - f_x(x_o,y_o+h)}{h} \tag{3.18}$$

Therefore, in partial derivative calculation at point (x_o, y_o), only variation in the desired direction must be applied. For example, for partial derivative in x direction, only variation in x direction is observed while y variable remains untouched.

From equation (3.8) it can be seen that voltage of the winding includes resistive voltage drop, inductive voltage drop and 'back-emf'. Equation is very similar to other motors, but one must have in mind that back-emf is calculated using partial derivative, and in contrast to other machines, it cannot be considered as the only component responsible for torque production [20]. In another words, product ei comprises the energy storing component and the conversion component, therefore it can not be used for torque calculation. Miller in [20] proposes to call this term *self-emf* instead of back-emf to emphasize that ei product doesn't comprise only conversion component. Equivalent per-phase circuit of equation (3.8) is shown in Fig. 3.35. Calculation of the parameters of the equivalent circuit has been demonstrated for rated conditions, but before apparent inductances have been calculated.

3.2.2.1 Calculation of Apparent Inductances

To calculate the apparent inductances of the given machine formula (3.3) was used. At the beginning only apparent inductances of the phase 1 were calculated. Simulation in which DC current of 12A was fed into phase 1 winding, while other windings were left open, i.e. with zero current, has been carried out. Speed of the machine was set to be 250rpm and it was simulated for a full electric cycle, i.e. 80ms. Flux linkages of all coils have been extracted and according to formula (3.3) apparent inductances $L_{i1}, i = \overline{1,6}$ have been calculated. They are presented in Fig. 3.36. It can be noticed that period of the inductance is 180° electrical. Therefore, in future simulations it is sufficient to simulate only half of the full electrical cycle and the rest of the waveform can be reconstructed in the post-processing via Matlab. This saves significant amount of time and memory resources. Moreover, new simulations for calculation of other inductances ($L_{i1}, L_{i2},$ etc.) are not necessary since inductances which share same diagonal in $[L]$ matrix are shifted for 30° electrical. In other words, $L_{ij}(\theta) = L_{i+1,j+1}(\theta + 30°)$. To prove this statement, another simulation was performed to calculate inductances $L_{i2}, i = \overline{1,6}$, where current of 12A DC was injected only in phase 2. Some comparison of apparent inductances L_{i1} and L_{i2} can be seen in Fig. 3.37-Fig 3.39.

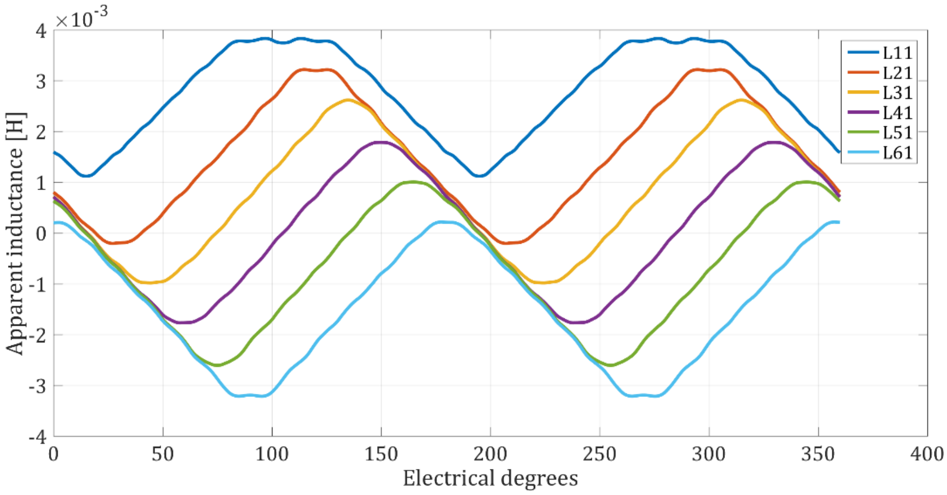


Fig. 3.36 Apparent inductances of phase 1

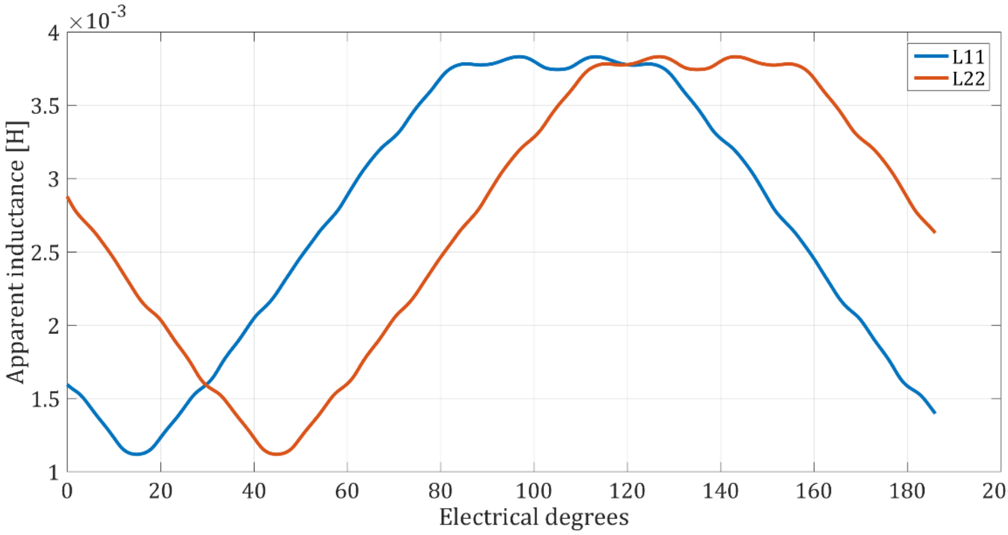


Fig 3.37 Apparent inductances L_{11} and L_{22}

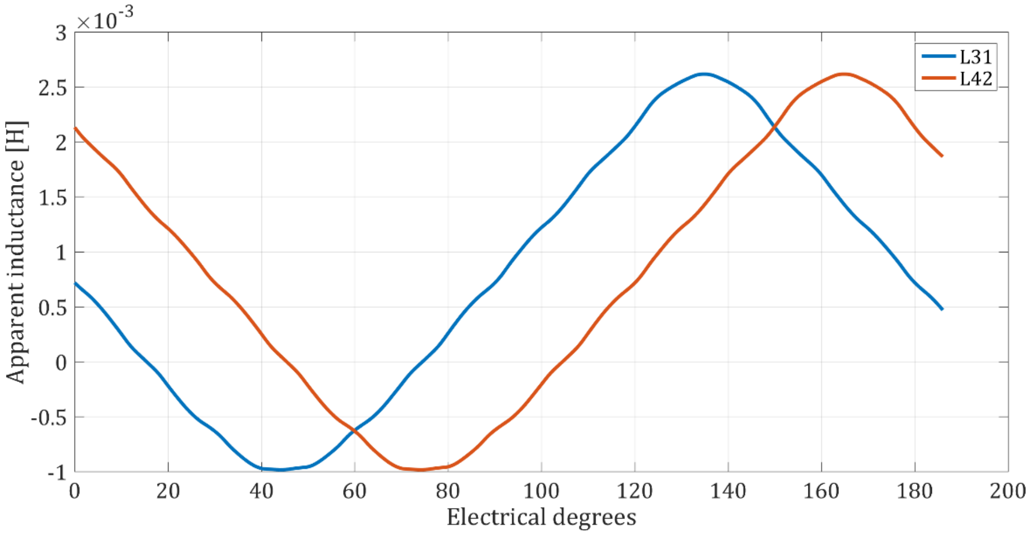


Fig 3.38 Apparent inductances L_{31} and L_{42}

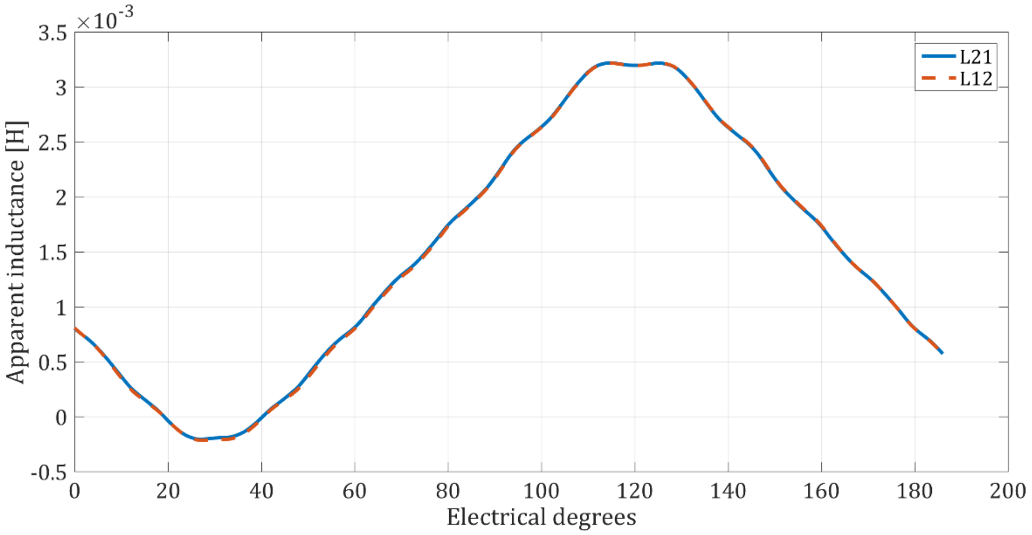


Fig 3.39 Apparent inductances L_{21} and L_{12}

What can be seen from these figures is that inductances which share same diagonals are shifted for 30 electrical degrees, while in Fig. 3.39 symmetry property ($L_{ij} = L_{ji}$) was demonstrated. Therefore, for calculation of $[L]$ matrix, only one column is needed while other inductances are calculated by shifting previously calculated inductances and using symmetry. In Fig. 3.40 all six self-inductances are shown. L_{11} and L_{22} are calculated using (3.3) while others are obtained by shifting its waveform in post-processing. All inductances are calculated for 1/6 of the machine, thus for full machine they must be multiplied with a factor of six.

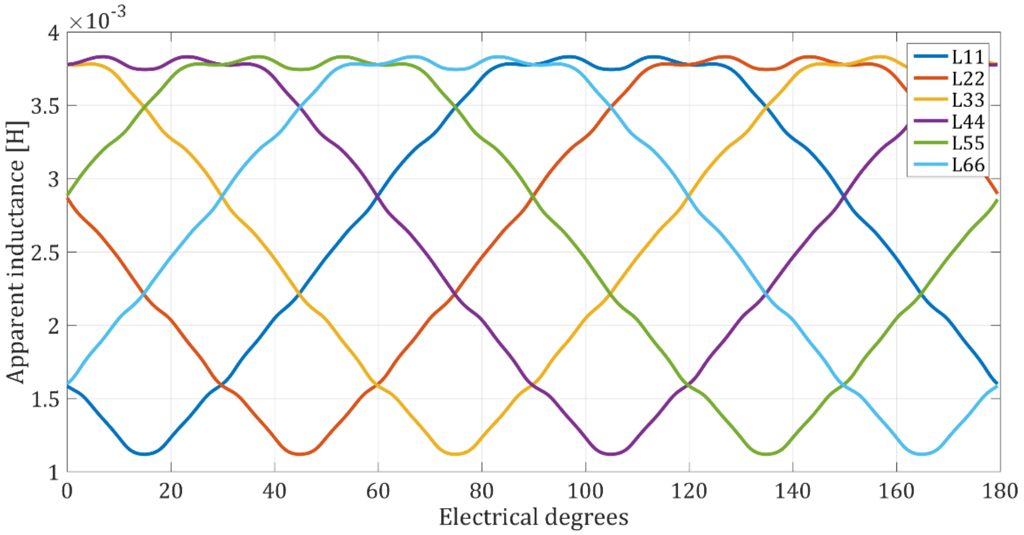


Fig. 3.40 Apparent self-inductances

3.2.2.2 Calculation of Incremental Inductances ($\frac{\partial \psi}{\partial i}$)

As explained before, in order to use equivalent circuit matrices $[L]$ and $[e]$ must be known in every instant of time. At the beginning $[L]$ and $[e]$ will be calculated for rated conditions. Considering previously mentioned facts it means that $\frac{\partial \psi}{\partial i}$ must be known in every time step of the simulation i.e. in every instant of time only variation (h) must be added only to the current of phase 1 while other currents and rotor position must remain the same. This doesn't mean that rotor must remain in the same position for all calculations as well as the currents of phases to 2 to 6. As explained in detail, all these partial derivatives are found in the neighbourhood of some operating point (x_o, y_o) but in every time instant new operation point is obtained. First incremental inductances will be calculated for rated condition and only $\frac{\partial \psi}{\partial i_1}$ have been calculated for the reasons stated in the section 3.1.2.1. Therefore, for calculation of this partial derivative two simulations are required. Both have rated currents, just one simulation is carried out with $i_{1\_plus} = i_{1\_rated} + h$ and the other with $i_{1_{minus}} = i_{1\_rated} - h$. Currents used in these simulations are plotted in Fig. 3.41 and in Fig. 3.42.

When time instant t_o is compared in these two simulations rotor position is the same (because time instant is the same), currents of phases 2 to 6 are the same in both simulations, except current 1. Clearly, flux linkages from these two simulations fulfil mathematical criteriums for partial derivatives (only difference in i_1). Incremental inductances are obtained using formula (3.14/3.15), and in this case it can be simplified and re-written as in (3.19)

$$L_{i1\_inc} = \frac{\psi_i(i_{1\_plus}, t_o) - \psi_i(i_{1_{minus}}, t_o)}{i_{1\_plus} - i_{1\_minus}} \tag{3.19}$$

Denominator, $i_{1\_plus} - i_{1_{minus}}$, in every instant of time is $2h$. According to [21] disturbance should be $2.5\% i_{rated}$, i.e. $h = 0.3A$. Hence, by performing these two simulations and extracting flux linkages, we can obtain first column of $[L]$ matrix. All other inductances are obtained by shifting 30° electrical as done in simulation with apparent inductances. Incremental inductances are shown in Fig. 3.43 and apparent L11 inductance and incremental L_{11} inductance are shown in Fig. 3.44. Also, all results are presented for 1/6 of the machine.

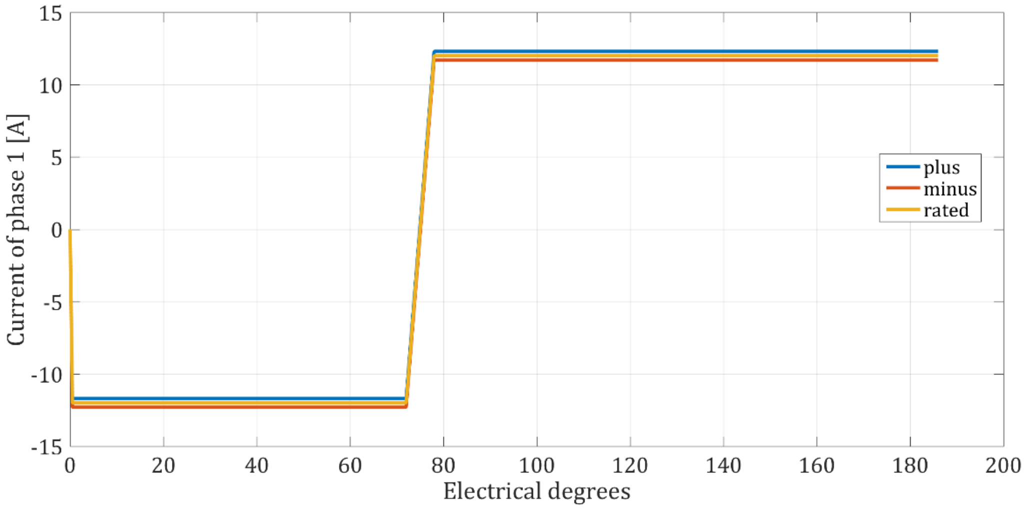


Fig. 3.41 Current of phase 1, rated and with disturbances

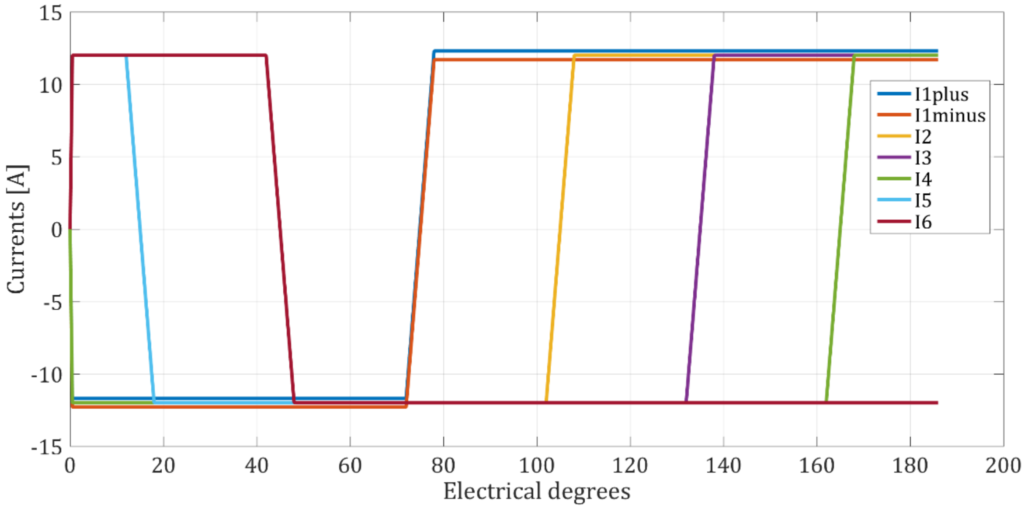


Fig. 3.42 Currents i_{1_plus}, $i_{1_{minus}}$ and rated currents of phases 2 to 6

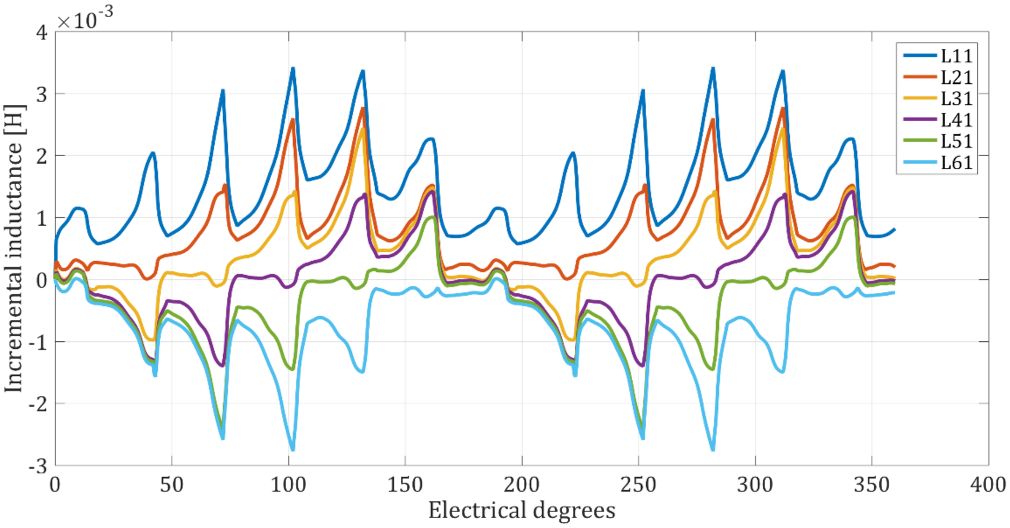


Fig. 3.43 Incremental inductances L_{i1}

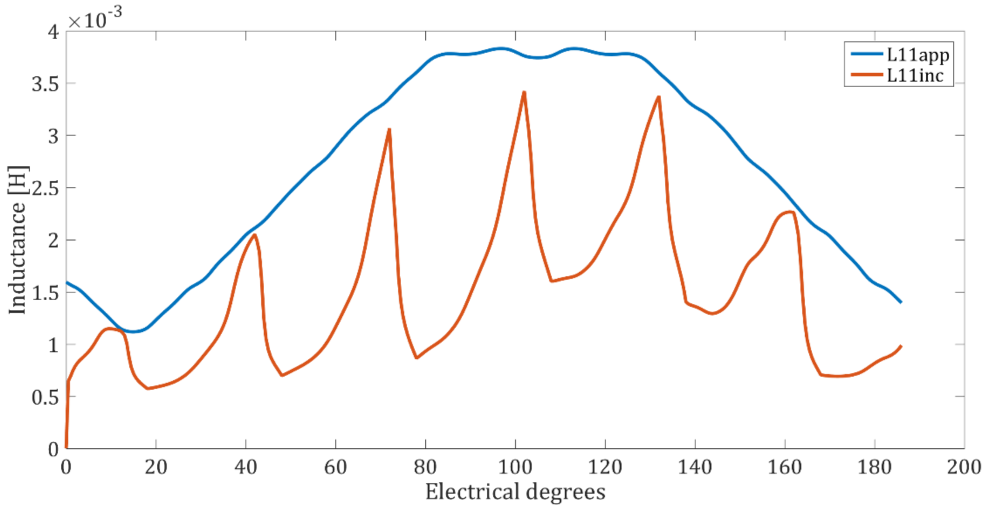


Fig. 3.44 Apparent and incremental self-inductance of phase 1

3.2.2.3 Calculation of Self-emf ($\frac{\partial\psi}{\partial\theta_r}$)

The goal of this simulation is to calculate $\frac{\partial\psi}{\partial\theta_r}$ in every time step of the simulation so $[e]$ matrix can be used in the equivalent circuit. Calculation of this partial derivative is more complicated than the previous one. According to the definition of partial derivative, to calculate this derivative in every instant of time, currents must be kept constant while rotor position must be varied. Since time and angle are connected by speed if we compare two consecutive time instants, like in calculation of $\frac{\partial\psi}{\partial i}$, position of the rotor is obviously different, but also currents are not always the same. Problem happens during commutation when in two consecutive time steps currents have two different values, instants t_3 and t_4 compared to instants t_1 and t_2 (Fig. 3.45). Therefore, a method to compute flux linkage in two different rotor positions with the same currents must be found. One solution is to have two simulations with shifted currents of 0.5 el degrees. If two consecutive time instants are compared but one from simulation with normal currents and other from simulation with shifted currents, for example t_1 and t_2 (Fig. 3.46), apparently currents are the same (approximately -10A) but obviously rotor position is different for 0.5 electrical in other words 1.5mechanical. Therefore, mathematical conditions to have same currents in two different rotor positions have been fulfilled. Although, we talked about partial derivative over rotor position in practice self-emf is calculated as follows

$$e = \frac{\psi_i(i_{1\_shifted}, t_o + \Delta t) - \psi_i(i_{1\_normal}, t_o)}{\Delta t} \tag{3.20}$$

where Δt is time step which corresponds to 0.5el degree. Obviously, division with angle and multiplication with speed is substituted only with division with Δt. Self-emf of phase 1 is shown in Fig. 3.47. while of phase 1 and 2 are shown in figure 3.48. Clearly, they are shifted by 30° electrical.

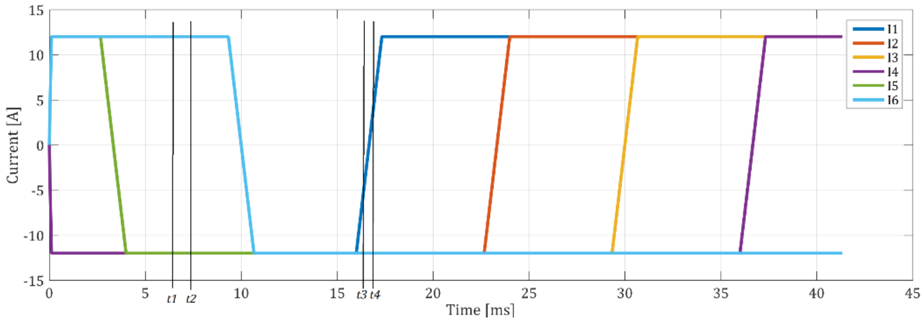


Fig. 3.45 Problem of having different values of currents in two consecutive time steps

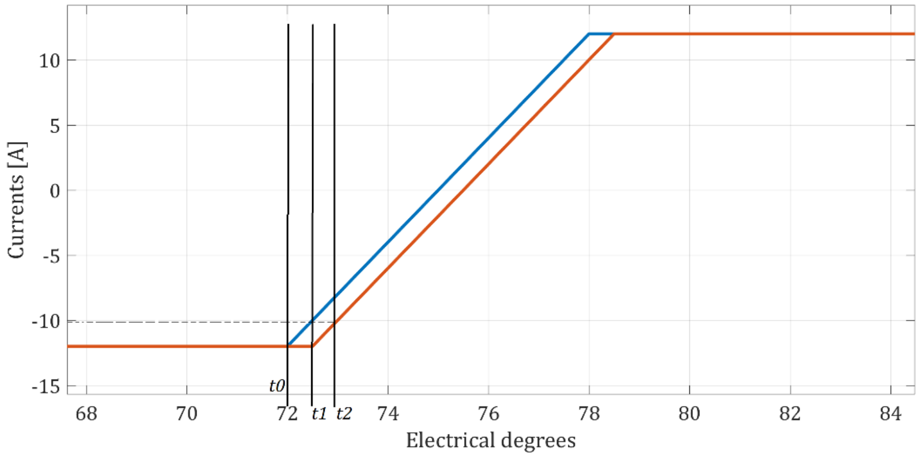


Fig 3.46 Solution of the problem with shifting current

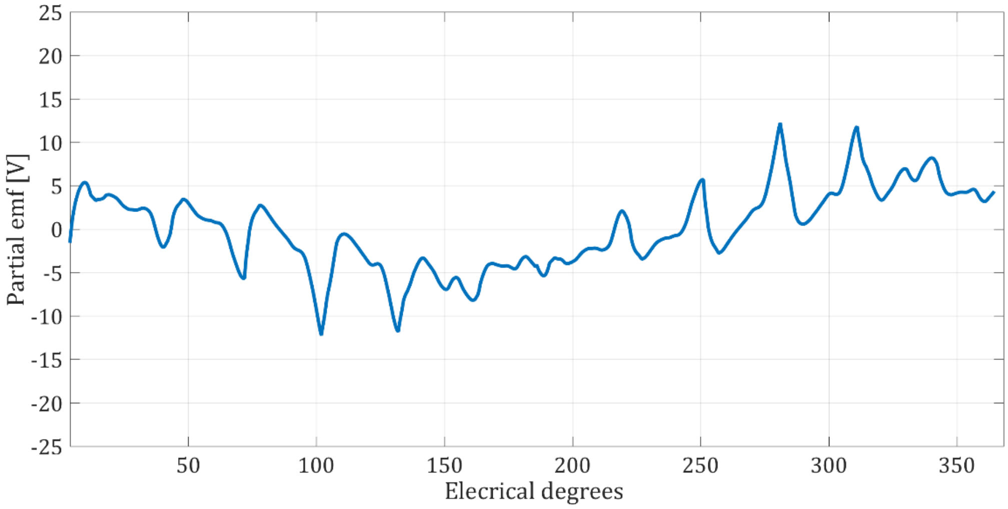


Fig. 3.47 Partial (self-emf) of phase 1

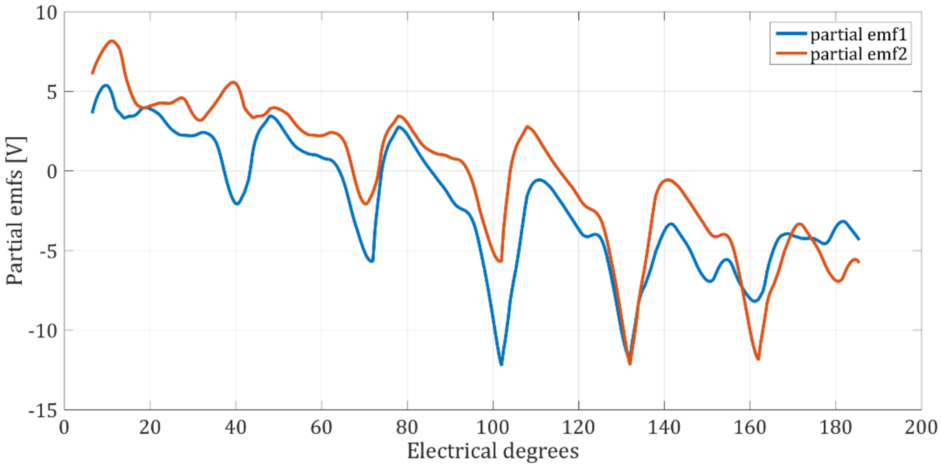


Fig. 3.48 Self-emfs of phase 1 and 2

3.2.2.4 Calculation of $\frac{d\psi}{dt}$

Calculation of $\frac{d\psi}{dt}$ is ordinary derivation in sense that nothing is kept constant. For this calculation, new simulation isn't needed. It is sufficient to use flux linkages from simulation of $(I_F, I_T) = (12{,}12)A$ presented in 3.1. Derivation in time instant t_0 can be calculated as in (3.21). If voltage drop over resistance is neglected then voltage over coil is being calculated. Voltage of coil 1 calculated using equation (3.21) is shown in Fig. 3.49. Moreover, voltage from MagNet is extracted and it is compared with $\frac{d\psi}{dt}$ in figure 3.50. Excellent agreement can be observed. Small delay which is noticeable can be caused by numerical differentiation.

$$\frac{d\psi}{dt} \approx \frac{\psi(t_0 + \Delta t) - \psi(t_0)}{\Delta t} \tag{3.21}$$

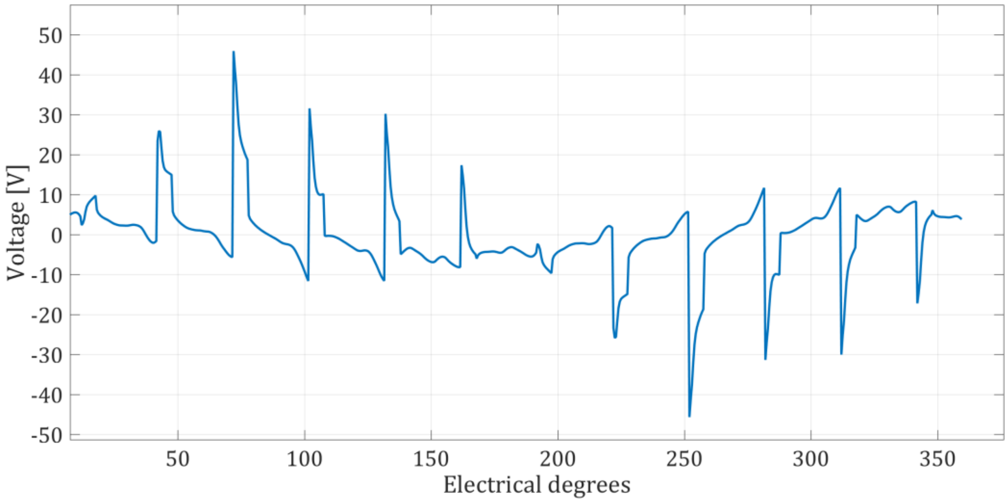


Fig. 3.49 Voltage of coil 1 calculated as $\frac{d\psi}{dt}$

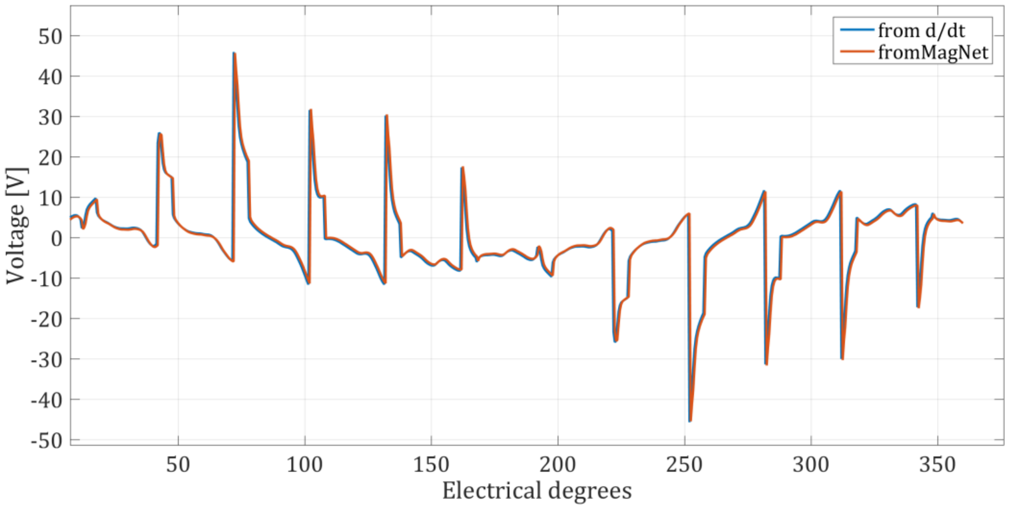


Fig. 3.50 Voltage from MagNet and calculated as $\frac{d\psi}{dt}$

At this moment, all necessary parameters for equation (3.7) have been calculated. Resistance is measured/estimated, self-emfs are calculated and matrix of incremental inductances has been obtained. Through post-processing in Matlab we can easily obtain derivative of the currents and therefore we can obtain voltage using (3.7). In Fig. 3.51 with blue line is shown voltage calculated using equivalent circuit and equation (3.6) while with red is shown voltage from MagNet.

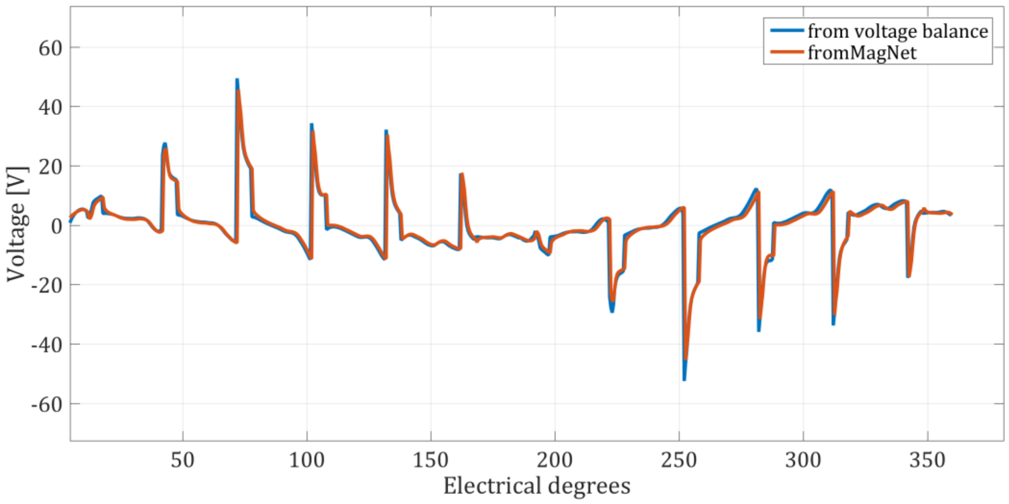


Fig. 3.51 MagNet and voltage calculated using parameters through equation (3.6)

Previous figure proves that self-emfs and incremental inductances are calculated correctly. Moreover, it proves that presented methods for calculation of partial derivatives are correct and that they can be used for finding parameters of equivalent circuit at any operating point. Again, it is worth to mention that all calculations have been performed for rated condition. In some other operating point these inductances cannot be used. Presented methods for incremental inductance calculation are also used to calculate incremental inductance in no-load condition to see the difference as it was done in [22]. Difference between load and no-load L_{11} incremental inductance is verified in Fig. 3.52. Another important fact which must not be forgotten is that self-emfs are not real back-emf since in product ei both conversion components and magnetic field are included. Therefore, product ei can not be used for finding torque like in the case of machines where linear model is used.

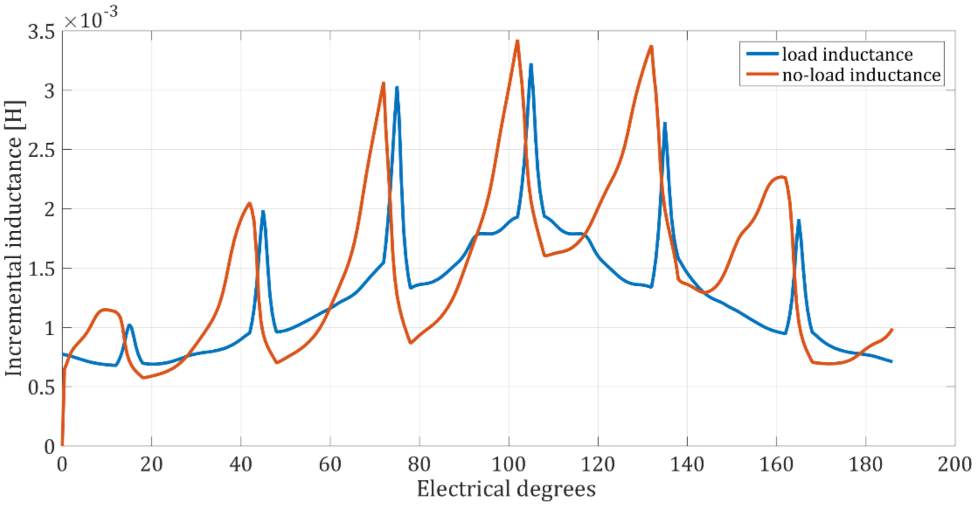


Fig.3.52 Incremental L_{11} inductance in load and no-load conditions

3.2.3 Torque Calculation Using Apparent Inductances

One way of torque calculation is to assume that machine is linear and to use linear inductances and formula (1.5). Formula (1.5) is derived from voltage balance equation. If both sides of equation (3.22) are multiplied with $[i]^T$ equation (3.18) is obtained considering that $[\psi] = [L][i]$. Left side presents electrical power $P_e = [i]^T[u]$ while $[i]^T[R][i]$ presents copper losses (P_{Cu}). Therefore, part $[i]^T\frac{d}{dt}([L][i])$ comprises power conversion and magnetic energy storage part.

$$[u] = [R][i] + \frac{d[\psi]}{dt} \tag{3.22}$$

$$[i]^T[u] = [i]^T[R][i] + [i]^T\frac{d}{dt}([L][i]) \tag{3.23}$$

$$P_e = P_{Cu} + [i]^T\frac{d[L]}{dt}[i] + [i]^T[L]\frac{d[i]}{dt} \tag{3.24}$$

Change of stored magnetic energy can be expressed according to [1] as:

$$\begin{aligned} P_{wm} = \frac{dW_m}{dt} = \frac{d}{dt}\left(\frac{1}{2}[i]^T[L][i]\right) = \\ = \frac{1}{2}\frac{d[i]^T}{dt}[L][i] + \frac{1}{2}[i]^T\frac{d[L]}{dt}[i] + \frac{1}{2}[i]^T[L]\frac{d[i]}{dt} \end{aligned} \tag{3.25}$$

Since P_{wm} is scalar $[P_{wm}] = [P_{wm}]^T$ and since $[L]$ is symmetric $[L] = [L]^T$, part $\frac{1}{2}\frac{d[i]^T}{dt}[L][i]$ can be written as:

$$\frac{1}{2}\frac{d[i]^T}{dt}[L][i] = \left(\frac{1}{2}\frac{d[i]^T}{dt}[L][i]\right)^T = \frac{1}{2}[i]^T[L]^T\frac{d[i]}{dt} = \frac{1}{2}[i]^T[L]\frac{d[i]}{dt} \tag{3.26}$$

Therefore, change of the stored magnetic energy is

$$P_{wm} = \frac{1}{2}[i]^T \frac{d[L]}{dt}[i] + [i]^T[L]\frac{d[i]}{dt} \tag{3.27}$$

Finally, right side of equation (3.24) can be re written as (3.28). From (3.28) power conversion part can be calculated and consequently torque as in (3.30)

$$P_e = P_{Cu} + \frac{1}{2}[i]^T \frac{d[L]}{dt}[i] + \frac{1}{2}[i]^T \frac{d[L]}{dt}[i] + [i]^T[L]\frac{d[i]}{dt} \tag{3.28}$$

$$P_e = P_{Cu} + P_{wm} + \frac{1}{2}[i]^T \frac{d[L]}{dt}[i] \tag{3.29}$$

$$P_C = \frac{1}{2}[i]^T \frac{d[L]}{dt}[i] = T\omega => T = \frac{1}{2}[i]^T \frac{d[L]}{d\theta}[i] \tag{3.30}$$

Torque according to this formula has been calculated for $I_F = 12$, $I_F = 6$ and $I_F = 2$. In this case new simulations aren't needed. Apparent inductances calculated in part 3.2.2.1 have been used (calculated with current of 12A) and in post processing in Matlab their derivative has been calculated. The part of Matlab code which describes calculation of one torque is presented below. Next to inductances there is index flip to indicate that they are obtained by shifting other inductances. `qrc` is time variable (180 electrical, 360 samples), and that is the reason why in code for torque calculation there is part `(360*3/(2*3.14))`.

```
dl11=diff(l11_flip)./diff(qrc);
dl21=diff(l21_flip)./diff(qrc);
dl31=diff(l31_flip)./diff(qrc);
dl41=diff(l41_flip)./diff(qrc);
dl51=diff(l51_flip)./diff(qrc);
dl61=diff(l61_flip)./diff(qrc);

dl12=diff(l12_flip)./diff(qrc);
dl22=diff(l22_flip)./diff(qrc);
dl32=diff(l32_flip)./diff(qrc);
dl42=diff(l42_flip)./diff(qrc);
dl52=diff(l52_flip)./diff(qrc);
dl62=diff(l62_flip)./diff(qrc);

dl13=diff(l13_flip)./diff(qrc);
dl23=diff(l23_flip)./diff(qrc);
dl33=diff(l33_flip)./diff(qrc);
dl43=diff(l43_flip)./diff(qrc);
dl53=diff(l53_flip)./diff(qrc);
dl63=diff(l63_flip)./diff(qrc);

dl14=diff(l14_flip)./diff(qrc);
dl24=diff(l24_flip)./diff(qrc);
dl34=diff(l34_flip)./diff(qrc);
dl44=diff(l44_flip)./diff(qrc);
dl54=diff(l54_flip)./diff(qrc);
dl64=diff(l64_flip)./diff(qrc);

dl15=diff(l15_flip)./diff(qrc);
dl25=diff(l25_flip)./diff(qrc);
dl35=diff(l35_flip)./diff(qrc);
dl45=diff(l45_flip)./diff(qrc);
dl55=diff(l55_flip)./diff(qrc);
dl65=diff(l65_flip)./diff(qrc);
```

```
dl16=diff(l16_flip)./diff(qrc);
dl26=diff(l26_flip)./diff(qrc);
dl36=diff(l36_flip)./diff(qrc);
dl46=diff(l46_flip)./diff(qrc);
dl56=diff(l56_flip)./diff(qrc);
dl66=diff(l66_flip)./diff(qrc);


TF12T12_lin=[];
I=[];
DL=[];

for i=1:359

I=[current1_f12_t12(i) current2_f12_t12(i) current3_f12_t12(i)
current4_f12_t12(i)  current5_f12_t12(i)  current6_f12_t12(i)  ];

DL=[dl11(i) dl12(i) dl13(i) dl14(i) dl15(i) dl16(i)
    dl21(i) dl22(i) dl23(i) dl24(i) dl25(i) dl26(i)
    dl31(i) dl32(i) dl33(i) dl34(i) dl35(i) dl36(i)
    dl41(i) dl42(i) dl43(i) dl44(i) dl45(i) dl46(i)
    dl51(i) dl52(i) dl53(i) dl54(i) dl55(i) dl55(i)
    dl61(i) dl62(i) dl63(i) dl64(i) dl65(i) dl66(i)
    ];

TF12T12_lin(i)=0.5*I*DL*I'*(360*3/(2*3.14));

End
```

Results are shown in Fig. 3.53 – 3.56. From figure 3.56. it can be seen that with $I_F = 12A$ usage of equation (1.5) gives significant errors for any torque current. This means that machine reaches non-linear operation point and that formula (1.5) cannot be used. With $I_F = 6$ results agree below $I_T = 10A$, while after 10A curves diverge one from another, for the same reasons as explained for the case of 12A field current. When field current is 2A formula (1.5) can be used with any torque current up to 12A. This difference, which is comparably lower than for field current level of 12A, might occur due to the use of apparent inductances calculated for 12A. Namely, apparent inductances, as discussed, is slope of the line connecting origin and (io,fo) and therefore, due to nonlinearity apparent inductances can differ if calculated for different current levels.

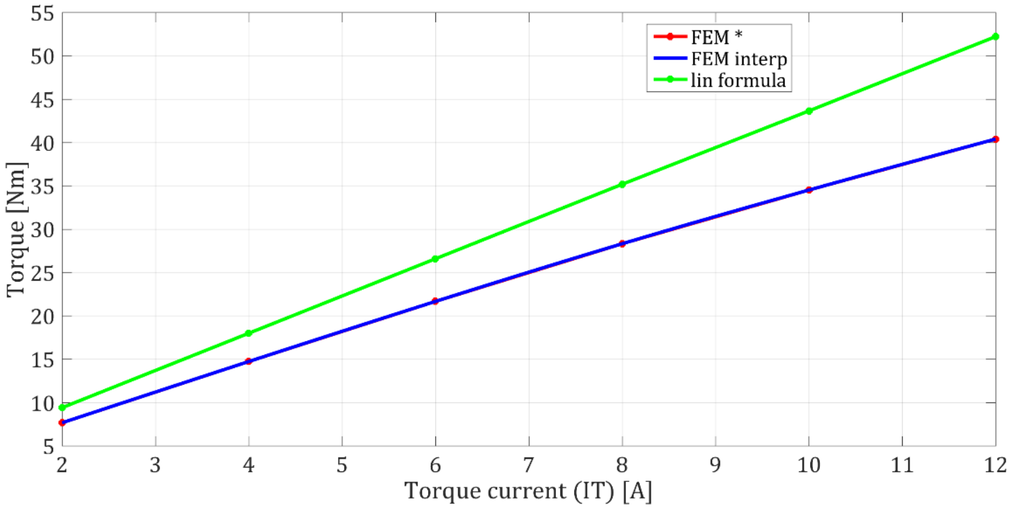


Fig. 3.53 Comparison of torque calculated with FEM and with linear formula with $I_F = 12A$

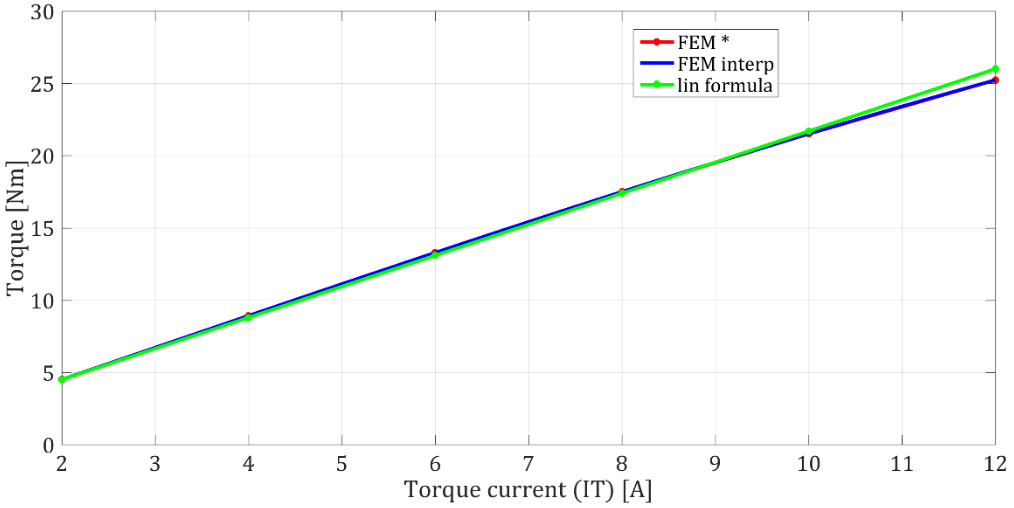


Fig. 3.54 Comparison of torque calculated with FEM and with linear formula with $I_F = 6A$

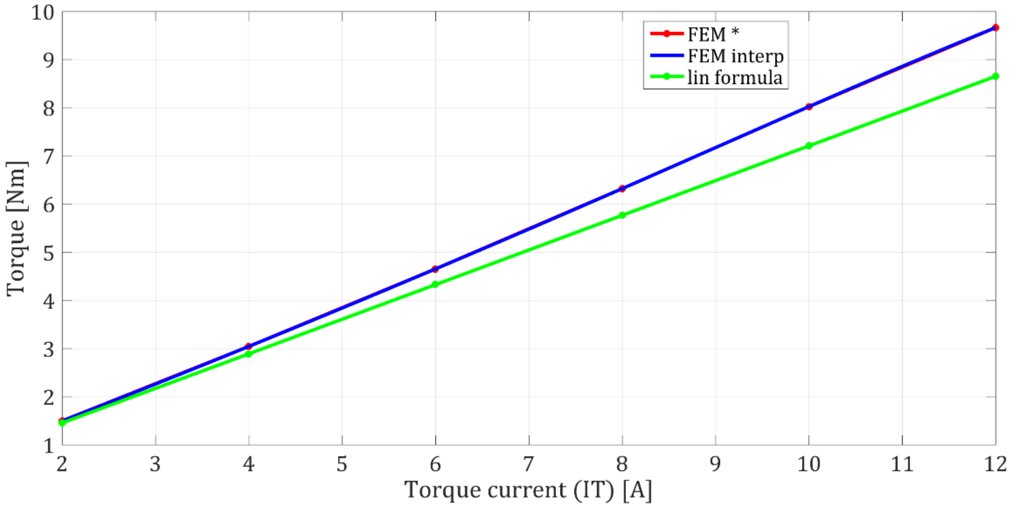


Fig 3.55 Comparison of torque calculated with FEM and with linear formula with $I_F = 2A$

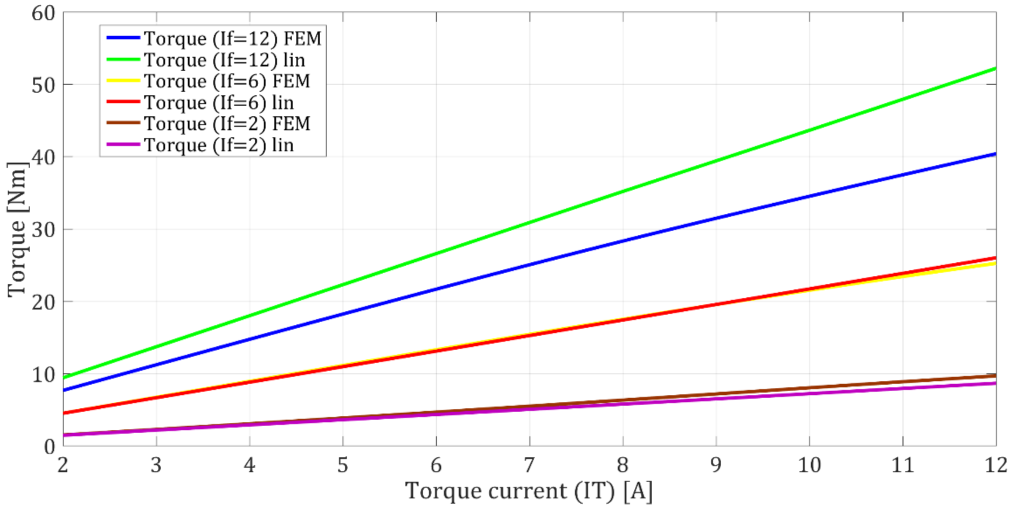


Fig. 3.56 Comparison of torque calculated with FEM and with linear formula for field currents of 12A,6A and 2A

3.3 Effect of The Air Gap Length

Following simulations ere done with different length of the air gap. Beside 0.35 *mm*, machine with air gaps of 0.6 *mm*, 1 *mm* and 1.5 *mm* were simulated with rated speed for various load and field currents $(I_F, I_T) = \{(12{,}12), (12{,}0), (6{,}6), (6{,}0)\}$. Machines with different air gaps are shown in Fig. 3.57 – Fig. 3.60.

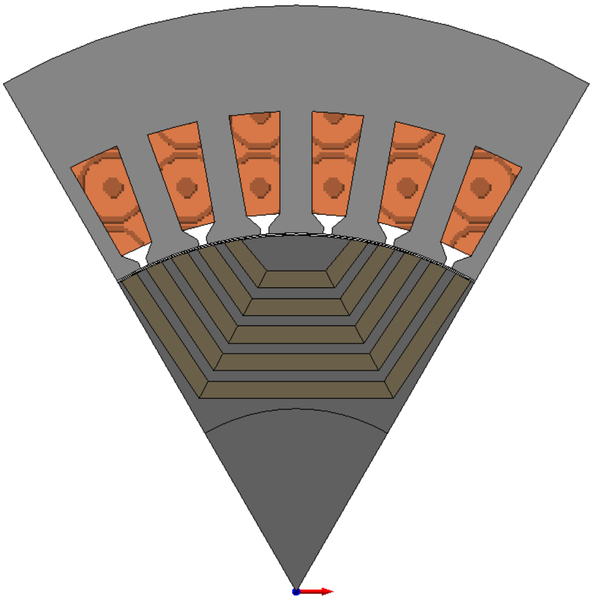


Fig. 3.57 Machine with 0.35 *mm* air gap

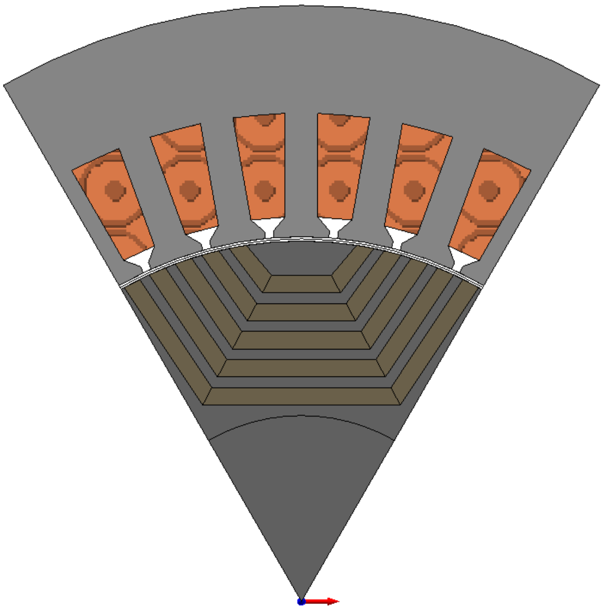


Fig. 3.58 Machine with 0.6 *mm* air gap

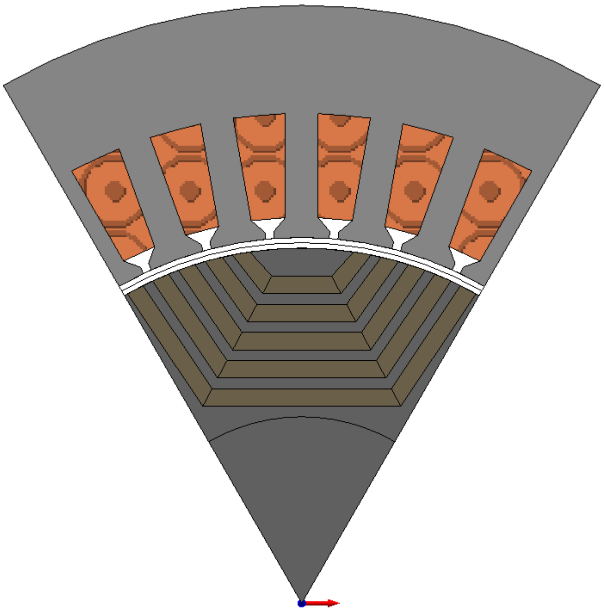


Fig. 3.59 Machine with 1 *mm* air gap

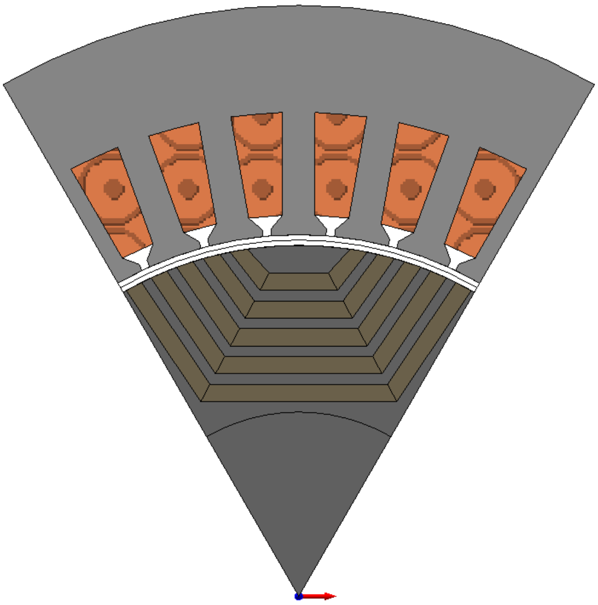


Fig. 3.60 Machine with 1.5 *mm* air gap

First analysis of no-load flux density will be done. Higher value of the air gap will increase reluctance. MMF is the same in all simulations since we have $(12{,}0)$ and $(6{,}0)$ for all four values of air gap. Therefore,

$$\mathcal{F}_{0.35} = \mathcal{F}_{0.6} = \mathcal{F}_{1} = \mathcal{F}_{1.5} \tag{3.31}$$

$$\mathcal{R}_{0.35} < \mathcal{R}_{0.6} < \mathcal{R}_{1} < \mathcal{R}_{1.5} \tag{3.32}$$

$$B = \frac{\mathcal{F}}{\mathcal{R}} => B_{0.35} > B_{0.6} > B_{1} > B_{1.5} \tag{3.33}$$

The conclusion is that for higher values of air gap we will need higher MMF for a given no load flux density. In the Fig. 3.61 and Fig. 3.62 it can be seen how average flux density decrease with air gap increase. Of course, that flux density for same air gap is higher for (12,0) then for (6,0). Flux density is extracted for time instant of 20ms for the middle of the air gap using arc graph probe for absolute normal smoothed B field (Fig. 3.63).

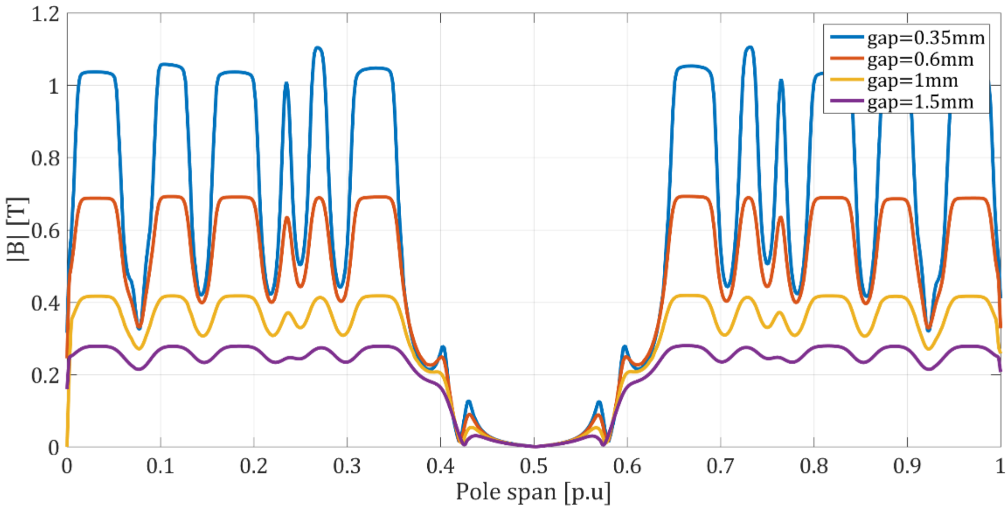


Fig. 3.61 No load flux density with $I_F = 12A$ and 4 different air gaps values

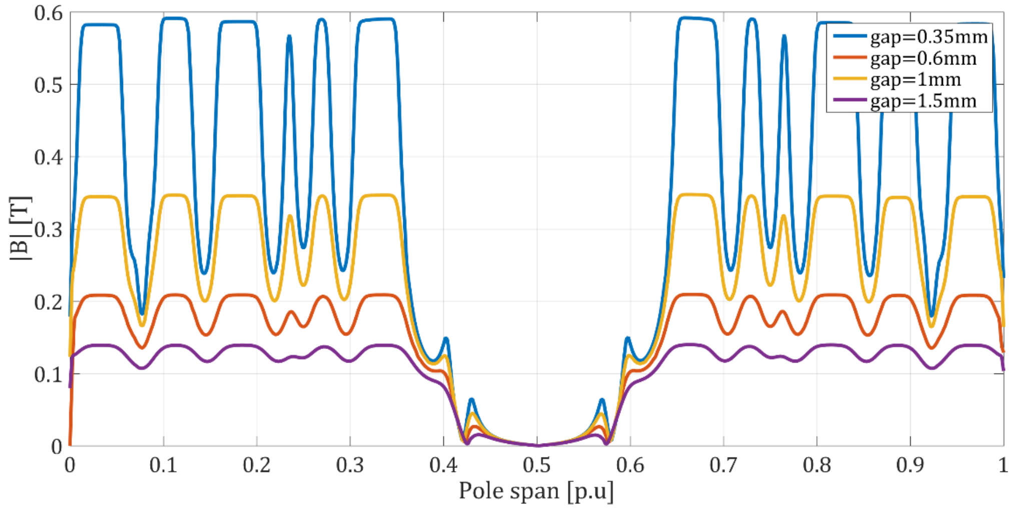


Fig. 3.62 No load flux density with $I_F = 6A$ and 4 different air gaps values

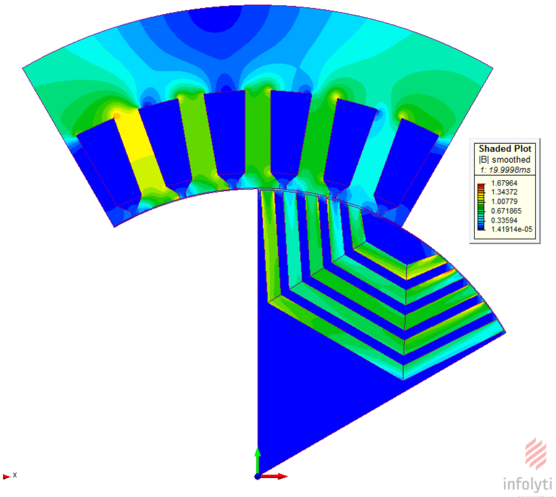


Fig. 3.63 Rotor position at 20ms

Second analysis is to see how increased air gap affects torque. It is expected to reduce it. Results (for full machine) are presented in Tables 3.5 and 3.6.

	T_{avg} $[Nm]$
Gap=0.35 mm	40.44
Gap=0.6 mm	30.5
Gap=1 mm	19.77
Gap=1.5 mm	13.02

Table 3.5 Average torque values for $I_F = 12A$ for 4 different values of the air gap

	T_{avg} $[Nm]$
Gap=0.35 mm	13.28
Gap=0.6 mm	8.29
Gap=1 mm	5.04
Gap=1.5 mm	3.26

Table 3.6 Average torque values for $I_F = 6A$ for 4 different values of the air gap

Regarding armature reaction, there isn't significant improvement with air gap length increase Fig. 3.64 and 3.65.

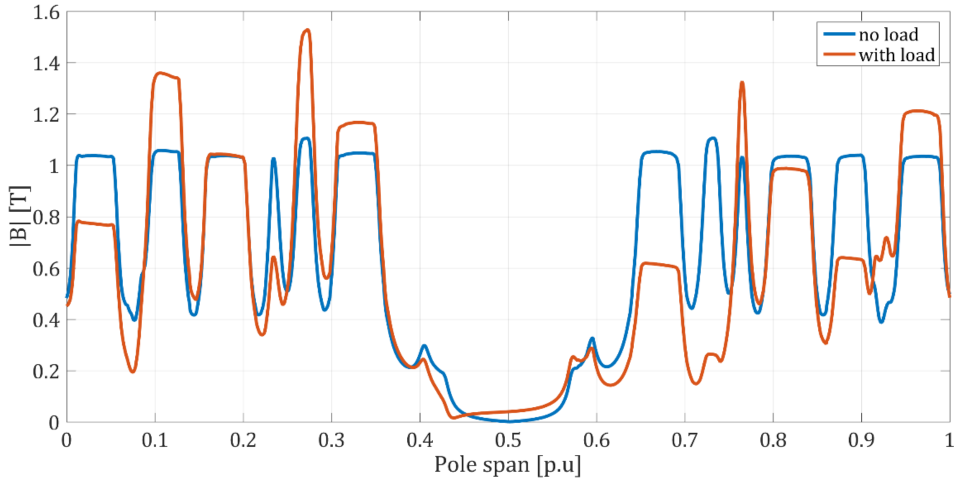


Fig. 3.64 No-load and full-load ($I_T = 12A$) flux densities with the air gap of 0.35 *mm*

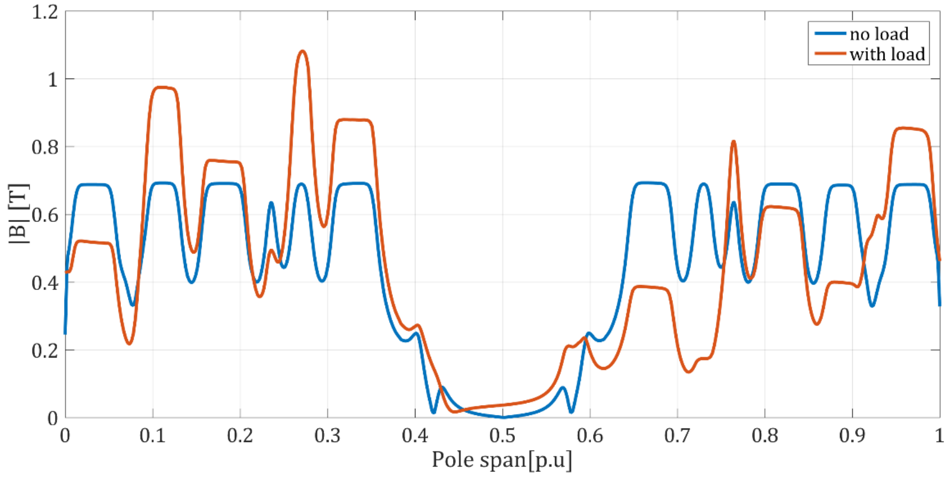


Fig. 3.65 No-load and full-load ($I_T = 12A$) flux densities with the air gap of 0.6 *mm*

3.4. Behaviour of The Machine versus Speed

In the following set of simulations speed is varied as well as torque current (6A and 12A) while field current is kept constant at 12A. Simulations are performed for speeds of 50rpm, 100rpm, 150rpm, 200rpm and 250rpm. Thus, in total 10 new simulations were performed. For these simulations, $\frac{W}{kg}$ curve was defined for rotor lamination. For every simulation mechanical (input) and electrical (output) power were calculated in post-processing by multiplying currents and voltages of corresponding phases and then summing them up. Knowing two previous powers efficiency of the machine is easily calculated as $\frac{P_{out}}{P_{in}}$. Results (for full machine) are presented in tables 3.7 and 3.8. For rated speed of 250 rpm efficiency of around 85% is expected. For lower values of currents, efficiency might be even higher due to lower copper losses in stator winding.

$(I_F, I_T) = (12,12)A$	$P_{mech(in)}$ [W]	$P_{el(out)}$ [W]	$\eta = \frac{P_{out}}{P_{in}} 100$ [%]
250rpm	1057.3	898.73	85
200rpm	846.43	697.14	82.36
150rpm	634.43	495.3	78.07
100rpm	422.66	293	69.32
50rpm	211.17	90.16	42.7

Table 3.7 Efficiency of the machine for $(I_F, I_T) = (12,12)A$ and different speeds

$(I_F, I_T) = (12,6)A$	$P_{mech(in)}$ [W]	$P_{el(out)}$ [W]	$\eta = \frac{P_{out}}{P_{in}} 100$ [%]
250rpm	566.21	486.33	85.9
200rpm	452.77	378.19	83.53
150rpm	339.42	269.94	79.53
100rpm	226.15	161.48	71.41
50rpm	113	52.8	46.73

Table 3.8 Efficiency of the machine for $(I_F, I_T) = (12,6)A$ and different speeds

In the following tables and graphs, losses which are dependent on speed are presented. They are iron losses in stator and rotor which are composed of two parts, hysteresis losses ($P_{hyst_{stator}/rotor}$) and eddy current losses ($P_{eddy_{stator}/rotor}$). Besides iron losses in stator and rotor, losses occur also in insulation between axial lamination and in spider, which are composed of stainless steel which has non-zero conductivity. Although, they are considered as ohmic losses they are induced by motion and hence they are speed dependent therefore they are also presented in tables 3.9 to 3.12. Results are shown for full-machine.

$(I_F, I_T) = (12,12)A$	P_{hyst_stator} [W]	P_{eddy_stator} [W]	P_{ohm_insul} [W]	Σ [W]
250rpm	14.58	4.04	11.92	30.54
200rpm	10.14	0.714	8.04	18.9
150rpm	7.32	0.402	4.81	12.532
100rpm	4.59	0.18	2.25	7.02
50rpm	2.076	0.045	0.59	2.711

Table 3.9 Stator hysteresis, eddy current losses and ohmic losses in insulation for $(I_F, I_T) = (12,12)A$ for various speeds

$(I_F, I_T) = (12,12)A$	P_{hyst_rotor} [W]	P_{eddy_rotor} [W]	Σ	$\frac{P_{Fe\_rot}}{P_{Fe\_stat}} 100\%$
250rpm	2.274	0.708	2.982	16.01
200rpm	1.734	0.453	2.187	20.15
150rpm	1.248	0.252	1.5	19.42
100rpm	0.786	0.1134	0.8994	18.84
50rpm	0.36	0.0286	0.3886	18.32

Table 3.10 Rotor hysteresis and eddy current losses for $(I_F, I_T) = (12,12)A$ for various speeds

$(I_F, I_T) = (12{,}6)A$	P_{hyst_stator} [W]	P_{eddy_stator} [W]	P_{ohm_insul} [W]	Σ
250rpm	8.94	0.585	5.51	15.035
200rpm	6.9	0.375	3.79	11.065
150rpm	4.97	0.211	2.29	7.471
100rpm	3.114	0.0942	1.09	4.3
50rpm	1.404	0.024	0.29	1.718

Table 3.11 Stator hysteresis, eddy current losses and ohmic losses in insulation for $(I_F, I_T) = (12{,}6)A$ for various speeds

$(I_F, I_T) = (12{,}6)A$	P_{hyst_rotor} [W]	P_{eddy_rotor} [W]	Σ	$\frac{P_{Fe\_rot}}{P_{Fe\_stat}}100\%$
250rpm	0.918	0.24	1.158	12.16
200rpm	0.714	0.153	0.867	11.91
150rpm	0.516	0.0864	0.6024	11.62
100rpm	0.324	0.0384	0.3624	11.29
50rpm	0.1464	0.0096	0.156	10.92

Table 3.12 Rotor hysteresis and eddy current losses for $(I_F, I_T) = (12{,}6)A$ for various speeds

It can be observed that rotor iron losses are in average 18.5% of stator iron losses for $I_F = 12A$ and around 11% for $I_F = 6A$. Total losses ($P_{hyst\_stator} + P_{eddy\_stator} + P_{Cu\_insul} + P_{hyst_{rotor}} + P_{hyst\_rotor}$) dependent on speed are presented in Fig. and Fig. As expected these losses decrease how speed decreases.

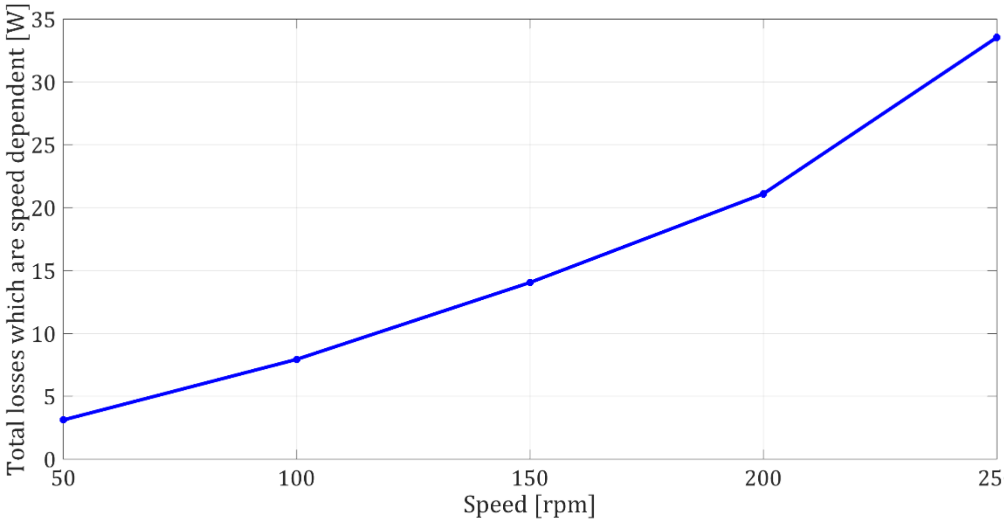


Fig. 3.66 Power losses ($P_{hyst\_stator} + P_{eddy\_stator} + P_{Cu\_insul} + P_{hyst_{rotor}} + P_{hyst\_rotor}$) as function of speed for $(I_F, I_T) = (12{,}12)A$

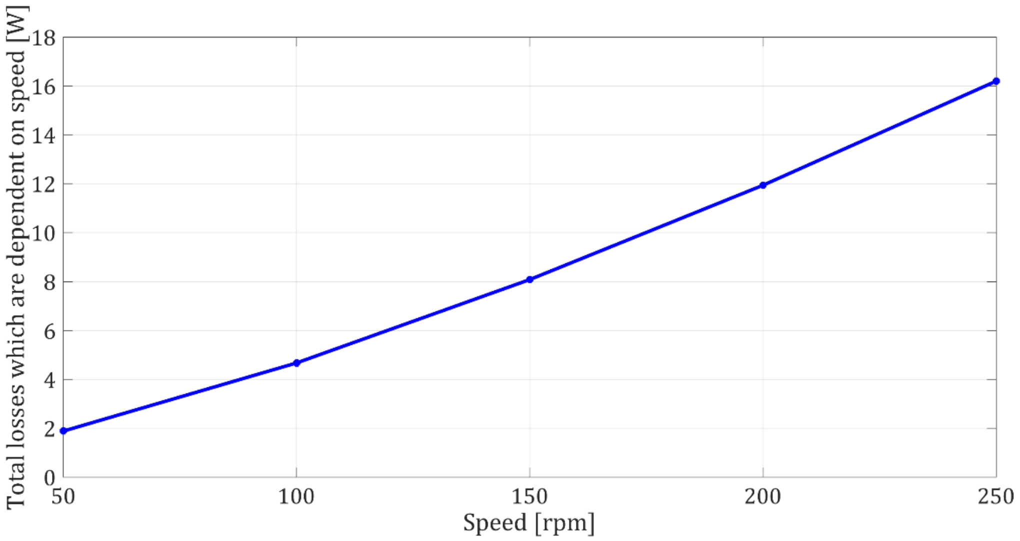


Fig. 3.67 Power losses $(P_{hyst\_stator} + P_{eddy\_stator} + P_{Cu\_insul} + P_{hyst_{rotor}} + P_{hyst\_rotor})$ as function of speed for $(I_F, I_T) = (12{,}6)A$

Chapter 4
Conclusions and Future Developments

4.1. Summary and Conclusions

In Chapter 1, the state of the art on multi-phase BLDC reluctance machines has been presented. The principle of operation has been explained as an electronically commutated brushless DC machine, with the armature moved to stator and the rotor made salient. The laminations of the rotor are placed axially, between which non-magnetic material is interleaved. Also a brief review of the main speed and torque control strategies have been presented in 1.3. Possible power converters for control of the currents in the windings have been explained as well as some drive configurations already built by two research groups.

Chapter 2, has been dedicated to the explanation of the modelling process. Only 1/6 of the machine has been modelled due to symmetry reasons as explained. MagNet program was used for modelling and simulation. However, the entire modelling has been automated through Matlab, i.e. a Matlab script has been created which communicates with MagNet and executes all modelling procedures from drawing of the edges to the assignment of the boundary conditions. In detail, the drawing process was explained step by step with figures presenting the model after each step. After successful drawing of the machine's edges, components were created by extrusion and material assignment. Special, anisotropic non-linear material was created which models the axially laminated rotor iron. Later, coils, currents through them and motion components were defined. For proper functioning of the model mesh and boundary conditions must be assigned, hence that has been presented in the last section of Chapter 2.

In Chapter 3, results of various simulations have been presented. All simulations are done with transient with motion solver to capture transients and power losses. The chapter is divided into four sections. In section 3.1 the global parameters of the machine such as torque, output power, voltage, efficiency and torque ripple have been presented in detail for rated conditions and for $(I_F, I_T) = (6{,}6)A$. Furthermore, torque and torque ripple have been investigated for another 34 pairs of field and torque current. Moreover, the effect of armature reaction was explored by comparison of load and no-load conditions. Section 3.2 was devoted to parameter identification. Resistance was measured experimentally and compared to the value obtained from the FEA model, while apparent and incremental inductances were calculated by performing specific FEA simulations. The methods and the mathematical background for calculation of incremental inductances and self-emfs for this type of the machine have been explained in detail. Also, it was verified that incremental inductances are parameters which are operating point dependent. In the end, it was shown up to which currents is possible to use the linear model of the machine. In section 3.3. the effect of air gap length has been investigated. Finally, in the last section, the machine was investigated for different speeds and various field and torque current to obtain power losses. Moreover, the efficiency was calculated for ten operating points and for instance, for $I_F = 12A$ and $I_T = 12A$ efficiency of 85% has been obtained with average torque of 40.38Nm and with output power of 898W, approximately 900W.

4.2. Future Developments

Even though the multi-phase BLDC reluctance machine is not new, not so much attention was devoted to it. Therefore, there are many possibilities for future exploration of this undoubtedly interesting machine. There are two potential directions for future developments. One direction is devoted to analysis of the machine itself. For instance, a more detailed parametric analysis of how the various geometric parameters affect the machine performances. With the help of Matlab script developed in this thesis, the change of these parameters doesn’t require preparation of a new model by hand and hence significantly saves the time. Other possible investigations are a) field weakening operation at high speed applications and b) fault analysis.

Another direction is towards application of this machine i.e. to the analysis of full electric dive. In particular, some future work can be devoted to power electronics and control algorithms. As presented in Chapter 1, not so much work has been done in this direction. At the beginning simulations could be performed and eventually test benches could be built. With the experimental testing of the machine new insights will be available, and new future applications can be found.

4.3. Quality Report

All work related to master thesis was done at Sapienza – University of Rome, Department of Electrical Engineering, which is one of the organisers of the EMMC STEPS programme. Developing thesis at this highly respected university had both personal and professional benefits for me. Working environment was satisfying and all programs necessary for thesis development were provided as well as the computer. Furthermore, thesis supervisor was daily available, his prompt replies even during the period of thesis writing are highly appreciated. Moreover, his advices were always helpful due to his eagerness and expertise in this topic. I have enjoyed working on this topic during which I gained valuable knowledge related to finite element analysis of the electrical machines and during which I deepened my knowledge about MagNet software.

Bibliography

[1] Slobodan N. Vukosavic, *Elektricne masine*, Akademska misao, 2010.

[2] Duane Hanselman, *Brushless Permanent Magnet Motor Design*, Magna Physics Publishing, 2006.

[3] Gianmario Pellegrino, Thomas M. Jahns, Nicola Bianchi, Wen L. Soong, Francesco Cupertino, *The Rediscovery of Synchronous Reluctance and Ferrite Permanent Magnet Motors*, Springer, 2016.

[4] H. Weh and U. Schroder, “Static Inverter concepts for multiphase machines with square-wave current-field distribution,” in Proc. EPE, Brussels, Belgium, 1985, pp. 1147–1152.

[5] A. E. Fitzgerald, C. Kingsley, S. D. Umans, Electric machinery, 6th edition, McGraw-Hill, 2003.

[6] Ion Boldea, Reluctance Synchronous Machines and Drives, Oxford Science Publications, 2012.

[7] Joseph D. Law, Modeling of Field Regulated Reluctance Machines, PhD thesis at University Wisconsin-Madison, 1991.

[8] J. D. Law, A. Chertok and T. A. Lipo, "Design and performance of field regulated reluctance machine," in IEEE Transactions on Industry Applications, vol. 30, no. 5, pp. 1185-1192, Sep/Oct 1994.

[9] E. T. Rakgati, M. J. Kamper and A. D. Le Roux, "Torque Performance of Optimally Designed Six-PhaseReluctance DC Machine," Conference Record of the 2006 IEEE Industry Applications Conference Forty-First IAS Annual Meeting, Tampa, FL, 2006, pp. 1186-1192.

[10] I. Boldea, L. N. Tutelea and D. Ursu, "BLDC multiphase reluctance machines for wide range applications: A revival attempt," 2012 15th International Power Electronics and Motion Control Conference (EPE/PEMC), Novi Sad, 2012, pp. LS1b.1-1-LS1b.1-6.

[11] D. Ursu, P. Shamsi, B. Fahimi and I. Boldea, "5 phase BLDC-MRM: Design, control, FEA and steady-state operation experiments," 2014 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), Bran, 2014, pp. 354-361.

[12] I. Boldea, Z. X. Fu and S. A. Nasar, "Performance evaluation of axially-laminated anisotropic (ALA) rotor reluctance synchronous motors," Conference Record of the 1992 IEEE Industry Applications Society Annual Meeting, Houston, TX, 1992, pp. 212-218 vol.1.

[13] D. Ursu, V. Gradinaru, B. Fahimi and I. Boldea, "Six-Phase BLDC Reluctance Machines: FEM-Based Characterization and Four-Quadrant Control," in IEEE Transactions on Industry Applications, vol. 51, no. 3, pp. 2105-2115, May-June 2015.

[14] S. Agarlita, D. Ursu, L. Tutelea, I. Boldea and B. Fahimi, "BLDC multiphase reluctance machines: A revival attempt with 2D FEM investigation and standstill tests," 2013 IEEE Energy Conversion Congress and Exposition, Denver, CO, 2013, pp. 1850-1857.

[15] MagNet introduction/ User guide/Tutorial

[16] MagNet online livedocs

[17] Nicola Bianchi, Electrical machines analysis using finite elements, CRC Press, Taylor and Francis group, 2010.

[18] James Stewart, *Multivariable Calculus*, 7th edition, Brooks/Cole Cengage Learning, 2012.

[19] A. Pouramin, R. Dutta, M. F. Rahman, J. E. Fletcher and D. Xiao, "A preliminary study of the effect of saturation and cross-magnetization on the inductances of a fractional-slot concentrated-wound interior PM synchronous machine," 2015 IEEE 11th International Conference on Power Electronics and Drive Systems, Sydney, NSW, 2015, pp. 828-833.

[20] T. J. E. Miler, *Brushless Permanent-Magnet and Reluctance Drives*, Oxford Science Publications, 1989.

[21] D. Ursu, L. Tutelea and I. Boldea, "Proposal with 2D FEM analysis of a six phase, 12 poles, 3kW, 200 rpm BLDC multiphase reluctance machine wind generator," 2013 15th European Conference on Power Electronics and Applications (EPE), Lille, 2013, pp. 1-9.

[22] E. T. Rakgati and M. J. Kamper, "Equivalent Circuit Parameter Identification of Six-Phase Reluctance DC Machine Using Finite Element Method," 2007 IEEE International Electric Machines & Drives Conference, Antalya, 2007, pp. 910-915.