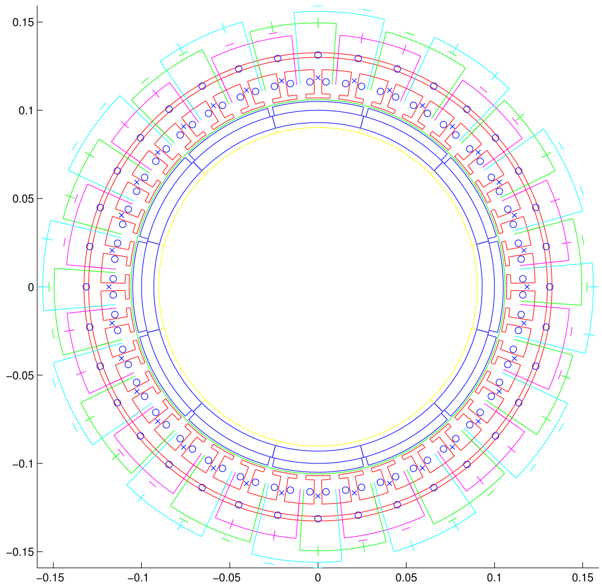

DC permanent magnet brushless boat motor for inside rotor mounted propeller

Master of Science Thesis

TOBIAS NILSSON

Department of Electrical Engineering
Division of Energy and Environment
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2007

DC permanent magnet brushless boat motor for inside rotor mounted propeller

Tobias Nilsson

Abstract

In this thesis a new type of boat motor is designed. The concept of the motor is to place the propeller inside the rotor, i.e. the motor is located around the propeller. Different options for motor type were considered. The final choice became the permanent magnet brushless DC motor. Different design parameters, number of slots, number of poles and more were considered as well as design options like sensor types. Magnetic material was evaluated with experiments in salt water environments. A program was written to make analytical calculations in Matlab. Finite element method was also done and program written to study the results.
Keywords: brushless, boat motor, magnets, salt water, sensors

Acknowledgements

I wish to thank my examiner and supervisor Dr. Sonja Tidblad Lundmark for providing valuable feedback on this work.

Contents

Chapter 1

Introduction

The aim of this thesis is to design a new type of electric boat motor. The idea is to make the motor and propeller to one compact unit. The propeller is placed inside the rotor. The rotor-propeller part is hold in place by a low friction material and lubricated by water. That gives a very compact design with very good cooling, since water flows between the rotor and stator. The power to the motor is supplied by batteries inside the boat. Also the control electronics will be located inside the boat.

Figure 1.1 shows an intended mounting on a boat. The figure is a cross-section from one side to the other of a boat at the point where the motor is mounted. It is mounted near the aft of the boat as ordinary propellers. In figure 1.2 the same mounting can be studying through an aft to fore cross-section. The stator part of the motor is mounted on a piece that can move inside a rectangular tunnel. During sailing or otherwise when the motor is not needed it can be lifted up in the rectangular tunnel, to lower the boats friction against the water. Ideally the motor is lifted above water level inside the tunnel to keep the motor out of the water to stay in better condition. To steer the boat the axis the motor is attached to is rotated by a steering motor. The ability to rotate the motor-propeller part gives the boat very good steering performance. The boat can rotate around its own axis. The electric motor together with the easily implemented steering equipment makes the boat easy to set up for automatic control at see and cable or wireless control for use in harbors.

The specific motor design in this thesis is intended to be mounted on the back of a 5 meters sailboat of a type called Stortriss. That way no hole in the bottom of the boat has to be made for testing of the new motor design. To be able to power a propeller suited for that boat type, the motor should be able to produce a torque of 20 Nm at 1000 rpm. The inside rotor diameter should be 18 cm to fit the propeller. There are no other dimensional restrictions so

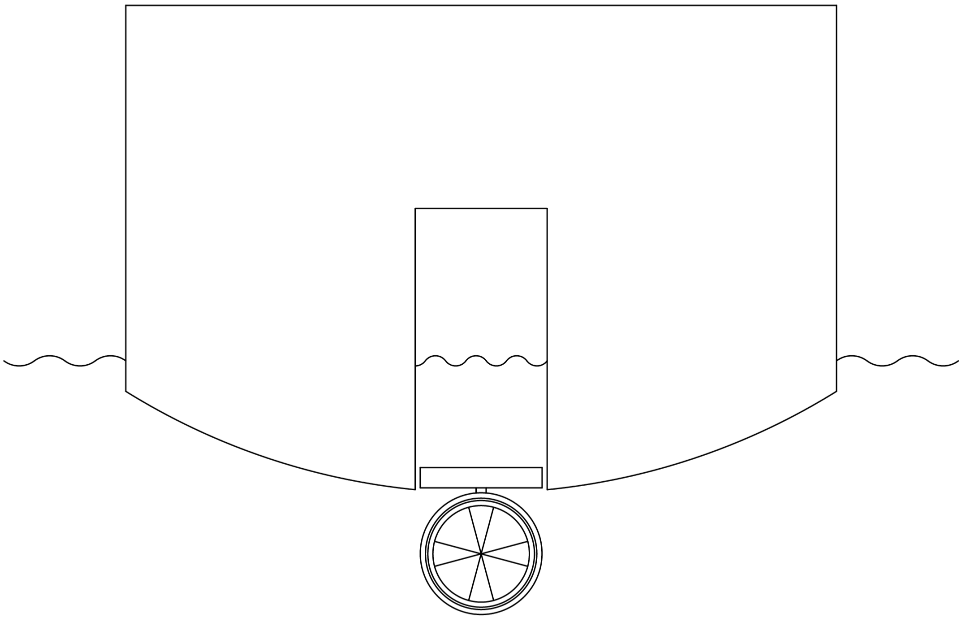

Figure 1.1: Cross-section of boat to see motor placement

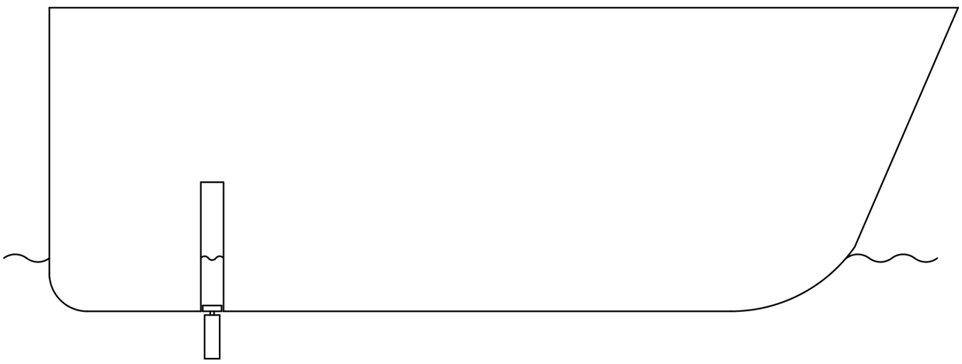

Figure 1.2: Cross-section of boat to see motor placement

the other dimensions can be chosen dependent of electrical and magnetically considerations only. When the voltage rating for the motor is to be chosen the fact that the motor will be powered by batteries have to be taken into account.

Another very interesting application of this type of motor is bow thrusters. A bow thruster of today can by seen in figure 1.3. A bow thruster is a propeller mounted in a pipe that goes through the front of a boat from one side to the other. The purpose of a bow thruster is to move water from one side to the other, in the front of the boat to make the front move sideways

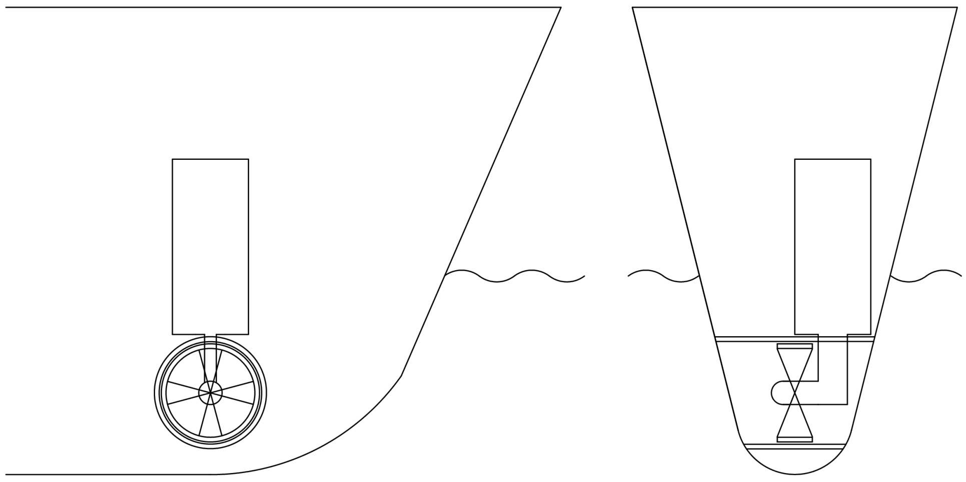

Figure 1.3: Design of bow thrusters today

in order to be able to maneuver with better precision. As can be seen in the figure the current design is to mount the propeller on a 90 degree gear and the electric motor is mounted above the pipe. The motor can take up a lot of space.

A simplified design example of the new type can be seen in figure 1.4. As can be seen the new design saves a lot of space.

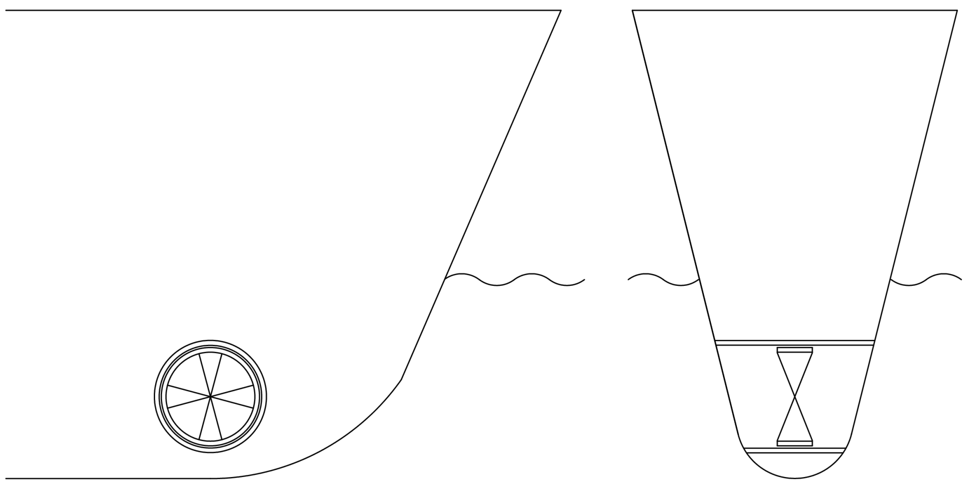

Figure 1.4: Design of bow thrusters with the new motor design

Chapter 2

Load characteristics

The motor in this thesis is designed for a type of sailing boat called Stortriss. A picture of that boat type can be seen in figure 2.1. It is about 5 meters long and weights 600 kg. Small boats are categorized in two main types. The

Figure 2.1: Stortriss

two types are called displacement and planing. The planing design is used for fast boats. The idea is to use high power motors and a flat bottom of the boat to make the boat go up from the water to decrease the resistance [1].

Since this motor is intended for a sailing boat, it falls under the displacement category. There are three main sources of forces the motor have to overcome. They are skin friction, form drag and wave drag. The skin friction is the friction between the water and the hull. Form drag is the force needed to move the water apart to make room for the boat in the water. The wave drag is the force that is lost then the boat makes waves[2]. The wave drag is small for speeds under the so called hull speed. The hull speed is calculated by[1]:

$$S = 1.34\sqrt{L_f} \tag{2.1}$$

Where L_f is the length of the boat at the waterline in feet and S is the hull speed in knots. With the length in meters, and S still in knots:

$$S = 2.4\sqrt{L} \tag{2.2}$$

For a Stortriss:

$$S = 2.4\sqrt{4.7m} = 5.2knots \tag{2.3}$$

For speeds above the hull speed a small increase in speed require a very big increase in motor power. The force depends of the sixths power of the speed[2]. For an electric motor design the maximum speed should be below the hull speed in order to save battery power. In addition to the motors power requirements dependencies of the speed of the boat, experiences also shows that the power requirements also depends heavily of the headwind speed (the wind speed facing the boat). When the wind speed in opposite direction of the boat is low, the boat can easily be powered by hand by rowing. On the other hand, at high head wind speeds, a 2 horsepower gasoline engine can only make the boat go very slowly forward. The power for the electric motor has been chosen to 2000 watts. That should be sufficient for powering the boat even at high headwind speeds. In most cases when driving the boat, the expected power need is however much below this maximum value.

Chapter 3

Motor type selection

Since there will be water between the stator and rotor the motor type must be one without brushes. Three types of motors were considered, the induction motor, the switched reluctance motor and the brushless DC motor.

3.1 Induction motors

Three phase induction motors are the most common in industry applications. Most of them are made to operate directly on the standard power net. The three phases are connected to coils in the stator. The windings are designed in order to produce a rotating magnetic field around the stator. This rotating field induces currents in the rotor windings, which are closed circuits. Since the stator field is rotating, a relative movement is obtained between the stator filed and the rotor coils that conduct the current. From that relative moment of the conducting coils in the stator filed a torque is produced.

If no load is attached to the motor the rotor will rotate with the same speed as the rotating magnetic field in the stator. When the load is increased the rotor will slow down. That makes more rotor winding pass by the stators field per time unit. That will in turn increase the current. Higher current in the rotor coils means higher torque. A drawback with the induction motor compared to the brushless DC motor is that there are losses in the rotor windings. These losses are actually needed for the motor to give torque [5]. Another drawback is that the motor needs three phase sinewaves in opposite to the brushless DC motor that only requires rectangular waves. This will increase the complexity of the control electronics and may increase the switching losses in the power components.

3.2 Switched reluctance motors

Switched reluctance motors have a stator with coils. The rotor is made of iron and doesn't have any coils nor magnets. The rotor have however slots and teeth. When two coils on opposite side of the stator are exited, the rotor wants to align rotor poles in a way so the magnetic flux have as much iron as possible on the way to the other exited pole. The torque obtained during the alignment is called reluctance torque. The design of switched reluctance motors is simple and robust in the sense that no magnets nor windings are needed in the rotor and only coils in the stator. The absence of magnets in the motor and the absence of windings in the rotor make the material cost low and the manufacturing cost low [7]. One drawback is that this type of motor is relatively new and the design procedure is complex [3]. Another drawback that makes it a bad choice in this design is that the air-gap have to be very small, about 0.2 mm [4]. That precision is difficult to obtain in a design like this.

3.3 Brushless permanent magnet DC motors

The brushless permanent magnet DC motor is basically an ordinary brush permanent magnet DC motor that have been turned inside out. That means that the stator magnets in an ordinary brush permanent magnet motor have been put in the rotor and the windings in an ordinary permanent magnets rotor have been put in the stator. That is the basic rotor-stator design. The brushless part of the name comes from that the brushes have been removed in the design and replaced by sensors and an electronic control circuit. The sensors detect where the rotor is located relative to the stator and control logic controls the currents to the stator windings so the correct windings are excited depending of the rotors position. The currents to the stator that the controller produce have rectangular waveforms.

Brushless permanent magnet DC motors have basically the same characteristics as their brushed friends but problems associated with mechanical commutations through brushes is removed. The drawbacks with brushless permanent magnet DC motors compared to induction motors is the need for sensors that makes the motor more mechanical sensitive and higher cost due to the permanent magnets.

3.4 Motor type chosen

From the discussions above the motor type chosen for this design was brushless permanent magnet motor. Is was chosen over induction motor mostly for the brushless motors higher efficiency that is a key issue in a battery powered design.

Chapter 4

Motor control

Brushless DC motors needs commutation circuits in order to work. The magnetic field in the stator have to rotate at the same speed as the rotor. Otherwise the rotor will stall and no torque is produced. In order to get the stator field to follow the rotor the commutation logic have to know where the rotor is. This is done with sensors or by analyzing the back EMF. When the controller know where the rotor is located, the controller can apply current in the correct phases.

The commutation logics only mission is to make sure the stator field follows the rotor movement. In other words, the fields speed isn't used to control the speed of the motor, like it is in for example stepper motors. The motors speed depends only of the magnitude of the current in the stator, in the same way a brush DC motors speed depends of the current in the rotor. The commutator logic only does what the brushes does in a brush DC motor. The brushless DC motor and the attached commutator logic together acts as a brush DC motor and the same power control by changing the current apply.

In addition to make the commutation circuits to get the stator field follow the rotor, the control circuits should also make sure the current don't rise over the maximum allowed value.

In addition to this low level control there can be a higher level control logic. The higher level control logic may control the motor to run at a certain speed or with a certain torque. That type of control is not needed here. The normal way for a human to control a boat motor is by controlling the power supplied to the motor. For a combustion engine that is done by controlling the fuel supply. For a combustion engine boat as well as for an electric motor boat the resulting speed or torque doesn't need to be accurate and constant. If the speed decrease as a result of for example an increased head wind speed the driver can adjust for that by increase the power supplied to the motor.

4.1 Power electronics

The power electronic base components for a three phase motor drive consist of a bridge with six transistors and six freewheeling diodes. At every moment current goes from one phase on the controllers output terminals, to the motor and back to the controller in another phase. The remaining phase don't carry any current at that moment. During a 360^o electrical interval each phase conducts positive current for 120^o, no current for 60^o, negative for 120^o and no current again for an interval of 60^o. As mentioned in the beginning of this chapter, brushless permanent motors speed is controlled by adjusting the current to the coils in the stator. This is done in the transistor-diode-bridge by pulse width modulation(PWM).

The current goes at every moment through one transistor in the controller, through one phase of the motor, back in another phase, and through another transistor of the controller. That means that the current always goes through two transistors. That brings up an issue of which transistor should be used for the PWM.

One strategy is to make the transistor for the phase that carries the positive current, i.e. the upper transistor, doing the PWM. The transistor for the phase that is carrying the corresponding negative current is conducting for the hole 120^o period without doing any PWM.

This strategy is however not optimal when taking the losses in the transistors in account. There are two types of losses to study, conducting losses and switching losses.

With conducting losses means the losses when a transistor conducts, i.e. $v \cdot i$ there i is the current through the transistor and v is the voltage drop over the transistor due to the internal resistance. i is usually big since it is the total current through the motor. v is small since the internal resistance in the transistor is small.

The other type of losses, the switching losses are due to the fact that the current through and the voltage over the transistor doesn't change momentarily then the transistor switches form off to on or from on to off. During the change there will be a short moment when there are quite a big voltage over and quite a big current through the transistor. The product of this current and voltage gives the switching losses.

In the strategy described above where one transistor does the PWM and one transistor conducts throughout the hole period, the conducting transistor only have conducting losses. The conducting transistor have switching losses for a short moment in the beginning of the period when it starts to conduct and in the end of the period when it stops to conduct. That is however negligible since it only happen twice for each period. On the other hand, the

PWM transistor will have less conducting losses but instead much switching losses since it is switching on and off throughout the hole period.

To make the transistors share the two types of losses equal, each transistor acts as a PWM during the first 60^o of it's conducting period. After that, it changes to conducting mode. That way the two types of losses will be equally distributed between the transistor pairs.

4.2 Position sensors

As stated before brushless permanent magnet motors needs to know in which angle the rotor is in order to apply current in the correct windings. There are a couple of different types of sensors that can be used to detect where the rotor is. Common types of sensors are [6]:

- Optical sensors
- Reluctance sensors
- Capacitance sensors
- Hall effect sensors (Magnetic field)
- Back emf sensors (Sensor less)

4.2.1 Optical sensors

Optical sensors consist of a disc attached to the rotor with a special pattern. The disc rotates between a light source and a light sensor which detects the pattern. The pattern can be made in different ways depending of the information needed. The simplest possible disc is one that only gives commutation information. A disc can have higher resolution if speed information at low speeds is needed. The most advanced design, is one with an absolute disc pattern, i.e. a pattern that gives information not only on where rotor magnets are located and the speed but also the exact position of the rotor. The additional speed and position information are useful in high precision closed loop control system. For a boat motor drive only the commutation information is needed.

The advantage with optical sensors in addition to the ability to give a lot of information as stated above is that the signals is very abrupt. The signal from the optical sensors goes form off to on very sharply.

Disadvantages with this type of sensors are that the light sources may fail and that a clean environment is needed. The need for a clean environment makes optical sensors a bad choice for this boat motor design. It's very likely that unclean water makes the sensor fail. Unclean water makes a direct obstacle for the light and my also make the disc dirty.

4.2.2 Reluctance sensors

Reluctance sensors consist of a wheel mounted on the rotor. The wheel have teeth with slots between. On the stationary part of the motor is mounted coils with approximately the same diameter as the teeth on the the wheel. Through each coil is an iron core. One end of the iron core faces the rotating wheel and the other end is connected to a magnet. Depending on if a tooth or a slot of the wheel is under the iron core the magnetic field through the iron core will be different. This difference can be measured with the coil and from that position information can be obtained.

One advantage with reluctance sensors is robust design. Disadvantages are need for signal processing in form of rectifying, filtering and triggering due to the gradually switching signal. In this design it's also a drawback that the rotor wheel will be very big since it need to have the same diameter as the rotor.

4.2.3 Capacitance sensors

Capacitance sensors have a wheel mounted on the rotor and sensors on the stationary part as the reluctance sensor have. However instead of measuring an induced current, rotor position is obtained by measuring the capacitance. The measured signal needs to be amplified a lot.

4.2.4 Hall effect sensors

Hall effect sensors use hall switches that are semiconductors that opens or closes depending of the magnitude of the magnetic field they are placed in. These hall switches are placed on the stationary part of the motor. They have to be placed carefully in order to not be disturbed by the stator field. The hall switches can detect the rotor magnets directly or magnets placed on the rotor only to be used as sensor magnets. Usually hall switches semiconductors contain circuits that produce a standard digital TTL signal, that switches on and off for certain threshold values of the magnetic field. A TTL signal is a digital signal with well defined voltage levels for indicating logical 0 and 1.

Advantages with Hall effect sensors are high accuracy and simple design.

4.2.5 Back emf sensors

Back emf sensors use the back emf in the stator windings generated by the rotor magnets. Both the back emf generated in the phase currently not in use (see section 4.1) in a three phase motor and the back emf in the two phases in use. Since no additional mechanical parts are added to the motor, this sensor type is also called sensor less. Because no additional mechanical components are needed there are nothing that can break. One drawback with back emf sensors is that noise in the phases due to the the switching electronics can interfere with the sensor logic and result in bad position sensing. Another drawback is that the back emf is very small at low speed so no position sensing is possible.

4.2.6 Sensor selection

Based on the advantages and disadvantages discussed above for different sensor types in this design hall effect sensors are chosen. These sensor will sense the rotor magnets directly. A thin layer of plastic can be added if the hall effect sensors need to be shielded for water.

Chapter 5

Design constants

In this chapter some design constants that is important in the design will be discussed.

5.1 Number of phases

Motors torque ripple decreases as the number of phases increases. The materials in the conductors are more utilized as the number of phases increases. A detail view of the torque and conductor utilization as function of number of phases can be seen in [8, page 3.5]. The drawback with an increasing number of phases is that more transistors and sensors are needed to generate the current to the motor. In addition more wires are needed between the power electronics and the motor. For example, a two phase motor, needs only two power switches and only one hall switch compared to a three phase motor that needs 3 sensors and 6 transistors. However in a two phase motor the torque can be zero at some positions and special arrangement have to be done in order to get the motor rotate in the desired direction. The two phase design is therefore only used in light-duty designs that have to be very cost effective, small fans for example. The most common trade off and the one that is done in this design is to use three phases.

5.2 Number of magnet poles

One advantage of having many magnet poles is that the torque the motor produce is directly proportional to the number of magnet poles. An increased number of magnet poles implies shorter pole pitch since more magnets have to fit in the same area of the rotor and therefore are closer together, this means shorter end-turns which leads to lower resistive power losses due to lower resistance, [9, page 97]. Additional benefits of an increased number of magnets poles are that for a doubling of the number of magnets, the rotor back iron and stator yokes thickness can be divided by two [8, page 3.6]. It is desirable to keep these thickness small in order to minimizing the area facing to water and in turn lower the water resistance.

One disadvantage with many magnet poles is that the flux leakage increases since there are more magnet ends [9, page 98].

Another important thing to study when choosing the number of magnet poles is the impact of the number of magnets pole on the relationship between the mechanical speed and the electrical speed. With electrical speeds means how fast the controller have to switch between the phases to the motor. It can be seen in the following equation that the commutation frequency is proportional to the number of magnets poles for a given mechanical speed:

$$\omega_e = \frac{N_m}{2}\omega_m \tag{5.1}$$

Where ω_e is the electrical frequency, ω_m mechanical frequency, N_m the number of magnet poles. In other words, an increased number of magnet poles requires an increased electrical frequency for a given mechanical speed. An increased electrical frequency have a couple of drawbacks in terms of power loss. This because a higher electrical frequency implies bigger losses in the switching electronics and core losses in the motor irons since the flux alternates direction at the same frequency as the commutation. The equation above gives that for permanent brushless motor design in general the number of magnet poles should be inversely proportional to the maximum speed of rotation. The mechanical speed on this motor should be 1000 rpm. The electrical frequency have been chosen to 100 Hz at nominal mechanical speed. A test with 50 Hz was done, but gave a substantial increase of the back irons. With 100 Hz electrical frequency the number of magnet poles will be calculated to:

$$N_m = 2\frac{\omega_e}{\omega_m} = 2\frac{2\pi 100}{1000/60 \cdot 2\pi} = 12 \tag{5.2}$$

5.3 Number of slots per pole per phase

The choice of the number of slots per pole per phase N_{spp} can affect the motor characteristics in a couple of ways. If the number of slots per pole per phase is chosen to a non integer value the cogging torque is usually lower. Cogging torque (reluctance torque) is an undesired torque, that comes from the effect that the rotor magnets wants to align to the stator so they are as close to as much stator iron as possible. A non integer value of N_{spp} will cancel the cogging torque of each end of the magnets so the total cogging torque is lower.

An advantage with a N_{spp} bigger than 1 is that the back emf will be more sinusoidal and smooth [9, page 112].

Having a higher N_{spp} has the disadvantage that the utilization of slot space usually is lower if there are many small slots compared to few bigger slots. In addition many slots consumes more work to wind [9, page 112].

Two number of slots per pole per phase was studied. The first was $N_{spp} = 1$ which gives a stator with 36 slots. The other design was $N_{spp} = 0.75$ which implies 27 slots. The winding approach for these setups will be presented in section 7.2. A comparison of the amplitude of the torque for the two setups were made. The result can be seen in figure 5.1. As can be seen the torque for the $N_{spp} = 0.75$ (27 slots) was lower than for the $N_{spp} = 1$ (36 slots). The torque ripple can be studied in figure 5.2. The torque ripple for the two designs where almost the same, therefore was $N_{spp} = 1$ (36 slots) chosen for the design.

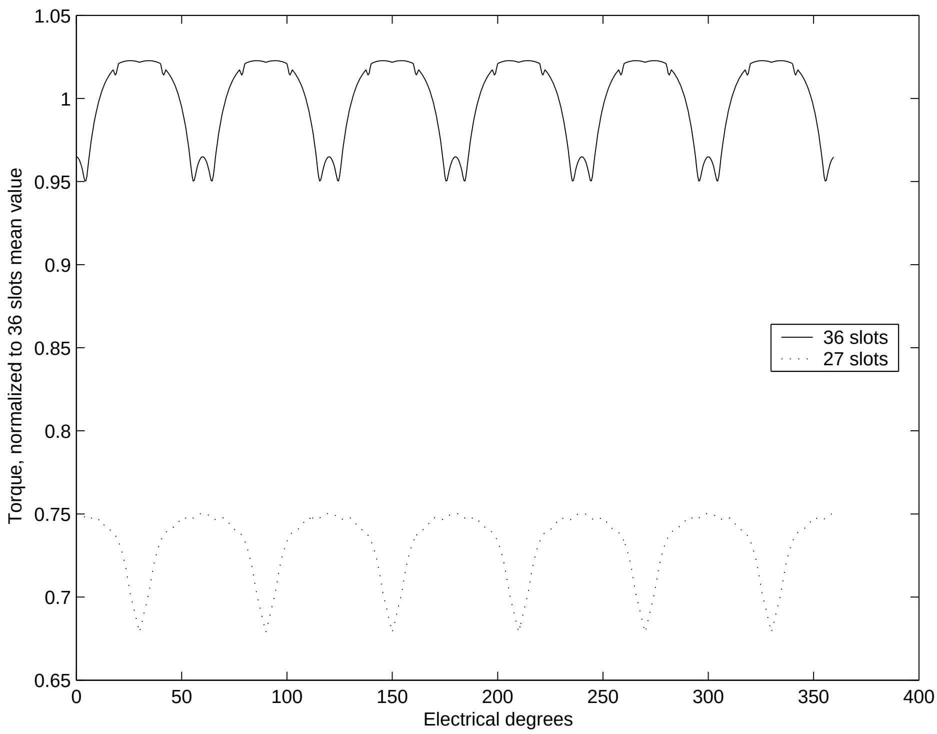

Figure 5.1: Comparison of the torque amplitude for a 36 slots design and a 27 slots design

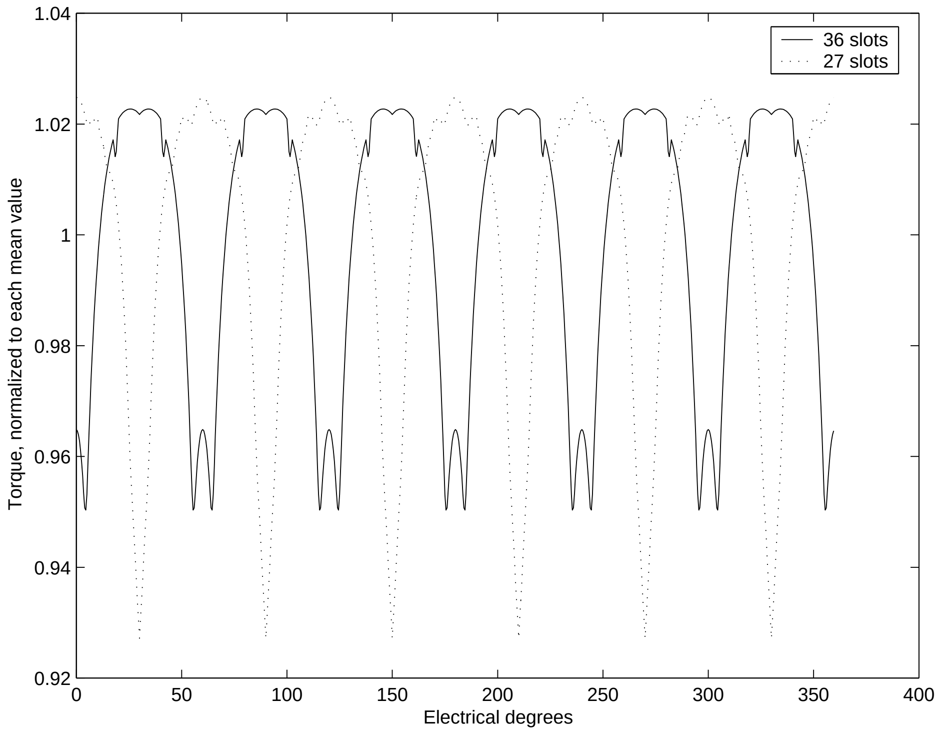

Figure 5.2: Comparison of the torque ripple for a 36 slots design and a 27 slots design

Chapter 6

Permanent magnets

Permanent magnets are fabricated by applying a high field to the material. Before the field is applied the material is unmagnetized, so the process start in the origin in figure 6.1. When the high field is applied the operating point moves along the line from origin up to the right. When the field is removed the operating point takes another way back. The way is illustrated with the upper curve in the same figure. As can be seen the other way don't end up in origin, meaning that there will be a remaining magnetization.

If no material are connecting the two poles of a permanent magnet (open circuit), the flux density B will be zero, (no current). The corresponding field intensity is denoted H_c. (Open circuit voltage). That point is marked in figure 6.1 at $(-\mu_0 H_c, 0)$.

On the other hand, if the two poles of the magnet is connected with an infinitely permeable material (zero resistance) H becomes zero (no voltage because of short circuit). The corresponding B is denoted B_r in the figure. This is the short circuit B and that is the highest B the magnet can provide.

During operation the permeance in the magnetic circuit connecting the poles is somewhere between zero and infinity. This implicate that the magnet will work between the points $(0, B_r)$ and $(-\mu_0 H_c, 0)$. These points are usually connected with a straight line having the slope $\mu_R \mu_0$. μ_R is called recoil permeability and is usually between 1.0 and 1.1. Near the $(-\mu_0 H_c, 0)$ point the line tend to bend sharply downward. If the magnet is forced to work beyond this point it will irreversibly decrease it's magnetization. To describe where the magnet is working a coefficient have been defined. It is called permanence coefficient, PC and is defined as the absolute value of the slope of a line from the working point to origin. The operating point in motors is usually chosen so the PC is of 4 or more [9, page 34]. It can be shown that for a permanent magnet in series with an air gap, the permeance coefficient

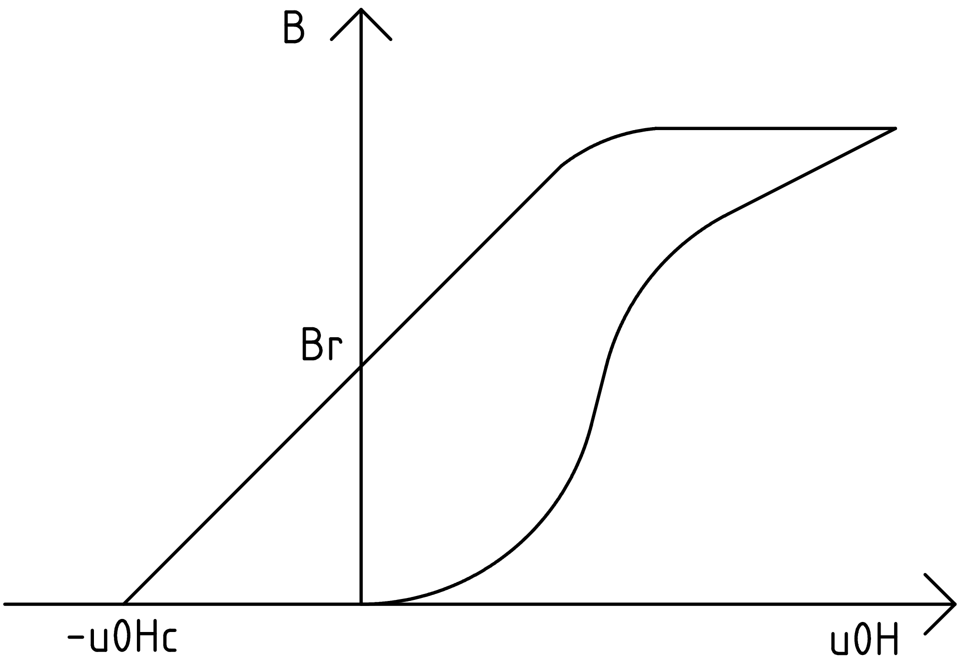

Figure 6.1: B-H curve for a permanent magnet. (u0H = $\mu_0 H$, -u0Hc = $-\mu_0 H_c$, Br = B_r)

becomes [9, page 38]:

$$P_c = \frac{l_m}{gC_\phi} \tag{6.1}$$

where C_ϕ is the flux concentration factor, l_m is the length of the magnet and g is the length of the air gap. The flux concentration is obtained by the design of the iron connecting the magnet with the air gap and the factor is calculated by:

$$C_\phi = \frac{A_m}{A_g} \tag{6.2}$$

where A_m is the magnet area facing the iron and A_g is the area of the iron facing the air gap.
A commonly used measure of magnet capacity is the $BH_{maxproduct}$ (current times voltage). It is the maximum product of B and H along the BH-curve.

6.1 Magnet materials

There are a couple of different magnetic material on the market with different properties regarding magnetic performance, temperature dependence, price, mechanical strength and oxidation. Temperature dependence is not expected to be an issue in this design due to the good cooling effect of the surrounding water. Mechanical strength is not a problem since a frame will hold the construction together. Oxidation is however an important issue in this design. There are four major types of magnets on the market:

- Alnico
- Ferrites
- Samarium-cobalt
- Neodymium

Alnico (Al-Ni-Co) have low magnetic performance but have good mechanical characteristics and they are heat resistant. They are used in motors working in a hot environment.

Ferrites (BaFe2O3 or SrFe2Or3) are the cheapest material and the most commonly used in motor constructions.

Samarium-cobalt (SmCo5 or Sm2Co17) have better heat Resistance and magnetic values then Ferrites. The drawback is that they are expensive and therefore only used in high performance motors.

Neodymium (NdFe14B) is the newest permanent magnet material and the one with the highest magnetic values. The working temperature should however not exceed $80C^0$, the price is a lot higher compared to Ferrites and they are corrosion sensitive.

6.2 Investigation of ferrite's oxidation properties

Since the motor will operate under water the oxidation properties of the magnetic material is interesting. The oxidation properties of ferrites, the cheapest type of material was tested. The test was done by placing a sample of the material at the Swedish west coast for half a year. The sample was hanged in a string just at the surface of the sea. The string was in turn attached to a iron beam a bit above the sample. Since the sample was just at the sea surface it was in a very corrosive environment since it was exposed to both salt water and air depending on the sea level at the moment and the whether in general. When the testing period had passed and the sample was picked up, it looked as can be seen in figure 6.2. At a first look it seemed like there had been some corrosion. After the first inspection the sample was cleaned with fresh water. After cleaning the sample looked as in figure 6.3. The stuff that looked like corrosion was easily removed with water. The rust on the sample must have come from the iron beam above. To summarize, the experiment gave the result that ferrite magnets can be placed in very corrosion intensive environment without being affected.

Figure 6.2: Corrosion sample directly from test

6.3 Choose of Magnetic material

Ferrites was chosen as magnet material. Neodymium was concerned since the working temperature is expected to be below $80C^0$ however the smaller dimension of magnets it gives don't make up for the higher price. The corrosion sensibility was also a big drawback in this design. The type of ferrite chosen was Y28 with $B_r = 370 - 400mT$ and Recoil permeability $mu_w = 1.05 - 1.3$. The mean of the intervals was used during the simulations.

Figure 6.3: Corrosion sample after cleaning

Chapter 7

Windings

First, a description of how the torque is produced from the interaction of windings and magnets. A simple setup is provided in figure 7.1. At the top of the figure a magnet and a rectangular coil layout is presented. The first magnet from left have a S-pole facing the coil. The second a N-pole and so on. This layout of magnets with alternating polarity follows to the left and right of the figure. The magnets are placed on a line instead of a circle. In addition return paths for the magnetic field are assumed to exist. This is for simplifying the understanding of the diagram but it illustrates in fact magnets on a circular motor.

The first diagram under the magnet-coil-layout is the flux density, (B) for the magnets. For the N-pole-magnet it's B_{max} and for the magnet with a facing S-pole it's $-B_{max}$. Next diagram illustrates the flux-linkage, (Ψ). The flux-linkage is obtained by multiplying the flux, (ϕ) through the coil by the number of turns of the coil, there the flux is given by B through the coil times the inside area of the coil:

$$\Psi = BA_{coil}N_{coil} = (\phi_{coil}N_{coil}) \tag{7.1}$$

The diagram shows the flux-linkage as the center of the coil moves from left to right over the three magnets. When the coils center is at the middle of the N-pole-magnet, as is the case in the figure, the maximum flux linkage, (FL_{max}) is obtained. Here all the flux from the N-pole-magnet goes through the coil. When the coils center is at the middle of a S-pole-magnet the flux-linkage reach it's maximum negative value i.e. the same amplitude as the maximum but with opposite sign, since the flux is negative for a S-pole-magnet. When the coils center is over the joint of a N and a S-pole magnet the flux linkage is zero since the flux from the two magnets cancel each other. Next diagram shows the back-EMF, (e). It is obtained by the Faraday's law, which states

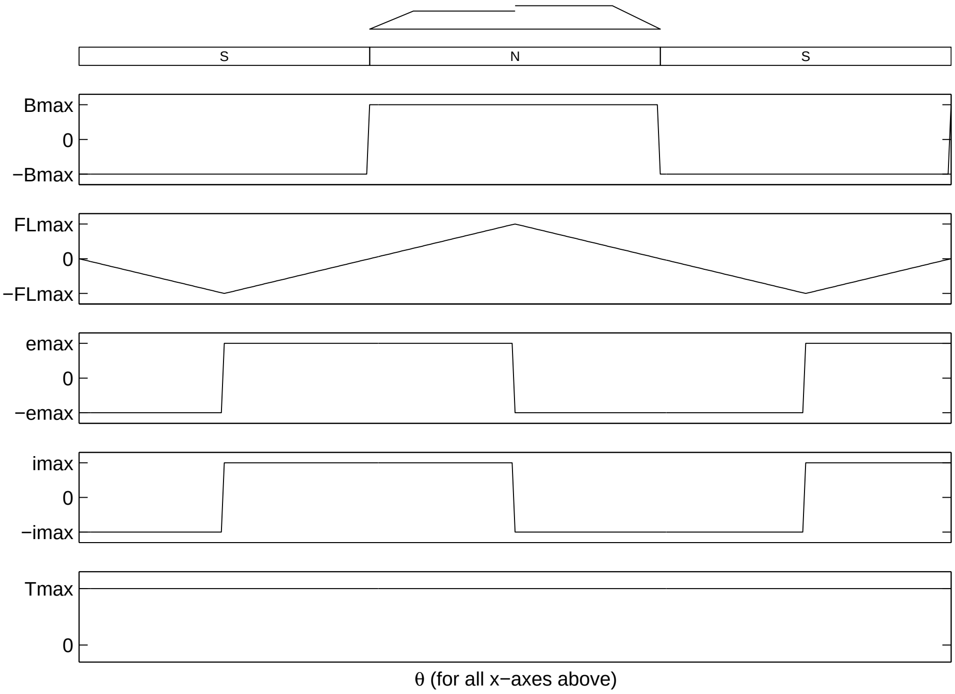

Figure 7.1: Waveforms from basic rectangular coil/magnet interaction when the coil at the top moves from one side to the other. More magnets follows to the left and right of the figure. (FL = Ψ)

that the back-EMF equals the changes of the flux-linkage. In math:

$$e = \frac{d\Psi}{dt} \tag{7.2}$$

Since it is assumed that the layout illustrates a circular motor, the time t above is not practical in the calculations. Instead t is replaced by use of the machines rotating speed, ω_m:

$$e = \frac{d\Psi}{dt} = \frac{d\Psi}{d\theta}\frac{d\theta}{dt} = \omega_m \frac{d\Psi}{d\theta} \tag{7.3}$$

The back-EMF is plotted in the third diagram. It have a constant positive value during the constant increase of the flux linkage. The constant decrease of the flux linkage gives a constant negative value of the back-EMF.

To produce torque, current is applied to the coil with the same polarity as the current back-EMF of the coil. The result is that a power of $i \cdot e$ is produced by the coil. The torque can be derived from the expression:

$$T\omega = ei \tag{7.4}$$

assuming no losses. The back-EMF is changing sign from time to another but the torque is in the same direction all the time by changing the direction of the current, whenever the back-EMF change sign. The current and the resulting torque is plotted in diagram four and five.

7.1 Winding approach

There are a lot of ways a motor can be wounded. One systematic algorithm will be presented here [8, page 3.39]. It will produce a double-layer winding. A double-layer winding is a winding where two coils goes in to every slot. The first coil starts by convention in slot zero or one. Slot one will be used here. Next, the maximum coil span will be calculated. This calculation gives the maximum coil span for this algorithm. However sometimes it can be better to choose a smaller coil span. That is called short-pitching or chording. Advantages with a coil-span less than the maximum is that it produce less harmonics and that the end-windings will be shorter. Shorter end-windings reduce the resistance losses. The disadvantages is an increased number of turns. The maximum coil span is given by:

$$\sigma_{max} = NLI\left(\frac{N_{slots}}{2p}\right) = \left[NLI\left(\frac{18}{2\cdot 4}\right) = 2\right] \tag{7.5}$$

The variables in the above equation are illustrated in figure 7.2. The numbers in the equation above between the sign "[" and "]" are also from the example setup in figure 7.2. σ_{max} is the maximum distance in units of slots, from the slot where a coil starts and to the slot where a coil ends. In other words the end-turns have to cross over σ_{max} number of stator teeth. In the figure the first coil is assumed to start in slot 1 so the σ_{max} in the figure, shows between which slots the first coil is located. NLI above is a function giving Next Lowest Integer, N_{slots} is the total number of slots and p is the number of rotor pole-pairs implying that $2p$ is the number of rotor poles (=number of magnets). Also N_{slots} and p is illustrated in the figure. Two things are known now. The first coil starts in slot one. It ends in the slot σ_{max} slots away, or less away if a shorter than maximum slot-pitch has been chosen. The next thing to obtain is where the next coil should start. The variable S_F gives how many slots forward from the previous coils end-slot the next coil should start. Also S_F is illustrated in 7.2. It is calculated by:

$$S_F = \begin{cases} N_{ss} - \sigma_{max}, & \text{if } \epsilon \leq 0.5 \\ \sigma_{max} + 1, & \text{if } \epsilon > 0.5 \end{cases} \tag{7.6}$$

ϵ and N_{ss} will be explained, ϵ first: If the number of slots divided by the number of poles in equation 7.5 for σ_{max} is an integer, the design is called an

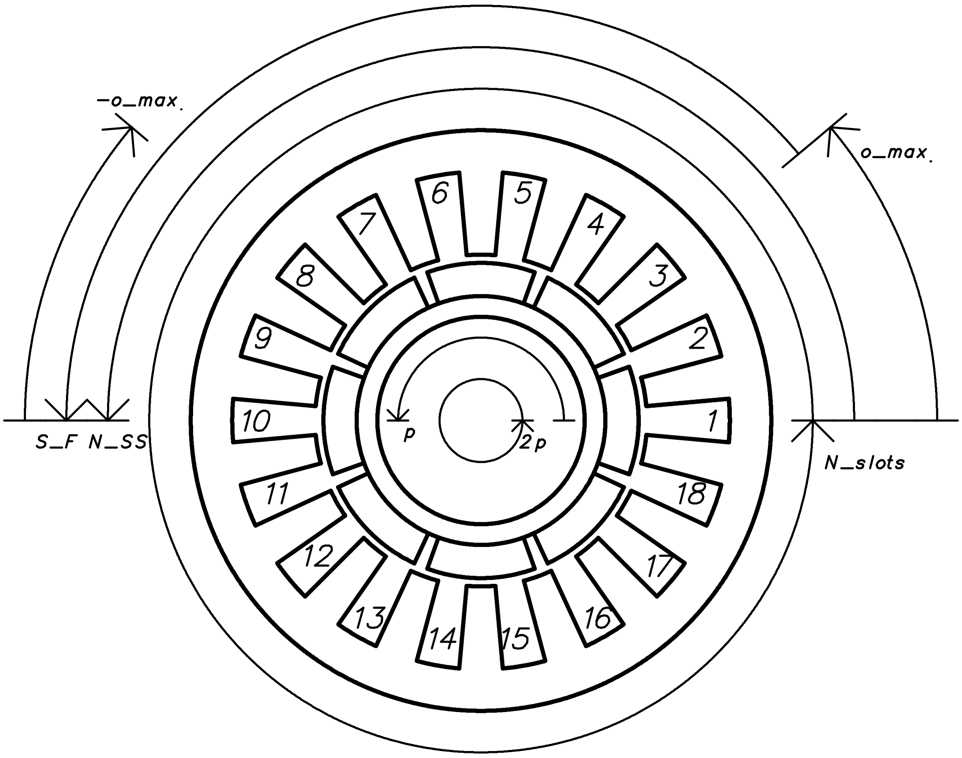

Figure 7.2: Illustration of some variables for the winding approach. (o_max $= \sigma_{max}$)

integral-slot design. If the division doesn't end up with an integer the design is called fraction-slot. In the case of fraction-slot the result from the division can be written as:

$$\frac{N_{slots}}{2p} = \sigma_{max} + \epsilon = \Big[2 + 0.25\Big] \tag{7.7}$$

where ϵ is the remainder (zero for an integral slot design). The numbers in the equation above between the sign "[" and "]" refers to the example in figure 7.2. The ϵ in this equation decides which case should be used in equation 7.6. Now to the other variable, N_{ss} in 7.6. To understand N_{ss} we first have to define a term called section. During the process of designing the windings the stator is thought of as being divided in sectors. Each sector contain S number of slots. It is given by:

$$S = HCF(C_{ph}, p) = \Big[HCF(6, 4) = 2\Big] \tag{7.8}$$

where HCF is the Highest Common Factor, p is the number of pole-pairs and C_{ph} is the number of coils per phase:

$$C_{ph} = \frac{N_{slots}}{N_{ph}} = \left[\frac{18}{3} = 6\right] \tag{7.9}$$

where N_{ph} is the number of phases. If C_{ph} and p don't have any bigger common factor than 1, the number of sections is 1. The variable N_{ss} is the number of slots in each sector. It is therefore given by:

$$N_{ss} = \frac{N_{slots}}{S} = \left[\frac{18}{2} = 9\right] \tag{7.10}$$

Also this variable is illustrated in figure 7.2.

The number of slots forward, S_F is now completely given by equation 7.6. In other words, now the start of the second coil is known. In the example it becomes $S_F = N_{SS} - \sigma_{max} = 9 - 2 = 7$

The polarity of the coils are altered for every coil. For example the return slot of the second coil will be in slot $-\sigma_{max}$ slots from the seconds coil's start slot. $-\sigma_{max}$ is inserted in the figure to illustrate this. Before proceeding, the placement of the two first coils will be summarized:

First coil

Start slot: The first coil starts by convention in slot 1.

End slot: The first coil ends in slot σ_{max} from the start slot. In the example: $1 + \sigma_{max} = 1 + 2 = 3$

Second coil

Start slot: The second coil starts in the first coil's end-slot plus S_F. With the first coil's end-slot from the previous calculation we get: $3 + S_F = 3 + 7 = 10$

End slot: The end slot is obtained by subtracting σ_{max} from the coils start slot. σ_{max} is subtracted instead of added since the coils should have alternating polarity. For the example: $10 - \sigma_{max} = 10 - 2 = 8$

Now when the two first coils has been summarized lets treat the third coil. The third coil starts S_F slots from the second coil's end-slot, exactly like the second coil started S_F slots from the first coil's end-slot. That implies for the example that the third coil should start at $8+S_F = 8+7 = 15$. The third coil's end-slot will be σ_{max} slots from it's start-slot, i.e. $15 + \sigma_{max} = 15 + 2 = 17$.

It is hopefully clear now, that a coil's start-slot is obtained by going S_F from previous coil's end-slot. The coil's end-slot is obtained by going alternating forward or backwards σ_{max} slots from the coil's start-slot.

This process will continue until the first sequence is done. The first sequence consist of the number of coils in the first sector for one phase:

$$C_{seq} = \frac{C_{ph}}{S} = \left\lceil \frac{6}{2} = 3 \right\rceil \tag{7.11}$$

It is called sequence since the coils don't have to be in the same sector, even if the number of coils in one sequence equals the number of coils in one sector. In the example the number of coils in one sequence is 3 as can be seen above. The third coil is therefore in first sequence but since the coil is in slots 15 and 17 it is not in the first sector.

If the number of sections S is one,equation 7.11 gives that $C_{seq} = C_{ph}$, which means that all the coils is in one sequence. In that case the winding for phase 1 is done then the first sequence is done.

Otherwise the remaining sequences have to be done. The second sequence is done by copying the coils in the first sequence N_{SS} coils forward. The second sequence will in turn be copied N_{SS} coils forward to make the third sequence. This continues until all C_{ph} coils are placed in the stator. In the example we have calculated the slots for the first 3 coil that fills the first sequence. Since the number of coils for each phase C_{ph} are 6 there are 3 remaining coils that all goes into the second sequence. The coil in the second sequence in the example will now be calculated by adding N_{SS} to the first coil's slots:

First coil in second sequence

Start slot: $1 + N_{SS} = 1 + 9 = 10$

End slot: $3 + N_{SS} = 3 + 9 = 12$

Second coil in second sequence

Start slot: $10 + N_{SS} = 10 + 9 = 19 = 1$ (slot 19 equals slot 1)

End slot: $8 + N_{SS} = 8 + 9 = 17$

Third coil in second sequence

Start slot: $15 + N_{SS} = 15 + 9 = 24 = 6$ (slot 24 equals slot 6)

End slot: $17 + N_{SS} = 17 + 9 = 26 = 8$ (slot 26 equals slot 8)

Remaining phases

The described procedure is for the first phase. The other phases have the same layout by shifted forward. The amount of slots each phase are shifted relative to the previous is given by the phase offset variable:

$$Offset = \frac{2}{3}\frac{N_{slots}}{2p} + k\frac{N_{slots}}{p}, k = 0, 1, 2 \tag{7.12}$$

Where the first k giving an integer offset should be used.

7.2 Windings for the motor

Two winding designs was analyzed. One for a 12 poles, 36 slots motor layout and one for a 12 poles, 27 slots layout. The second is a fractional slot and was studied in order to see if the cogging torque was lower.

7.2.1 Winding for 12 poles, 36 slots

The first winding is for a design with 12 poles (6 pole pairs) and 36 slots. Since number of slots per pole per phase is an integer $36/12/3 = 1$, the design is an integral slot winding. As mentioned earlier the described winding procedure will produce a double layer winding so this winding will be a double layer. The first variable to calculate is the maximum coil span:

$$\sigma_{max} = NLI\left(\frac{N_{slots}}{2p}\right) = NLI\left(\frac{36}{2 \cdot 6}\right) = 3 \tag{7.13}$$

In this design the coil span has been chosen to the maximum possible according to the equation above. If a smaller than maximum coil span had been chosen, for example the next lower, 2, then the coil span would have been less than 2/3 of the distance between two magnet centers. That can be seen by looking at figure 7.3. So small coil spans are usually not used.

With a coil span of 3, the first coil will start in slot 1 and end in slot $1 + 3 = 4$. This first coil can be seen in figure 7.3. In the figure slot 1 is found to the right just above the center of the motor. The arrow on the coil indicates the polarity. Next thing to calculate is ϵ to decide which case to use in equation 7.6:

$$\frac{N_{slots}}{2p} = \frac{36}{12} = \sigma_{max} + \epsilon = 3 + 0 \tag{7.14}$$

Since the remainder ϵ is zero, the first case in equation 7.6 should be used. The number of slots per section, N_{SS} therefore have to be calculated. To do that, the number of coils per phase and the number of sections have to be calculated first. Number of coils per phase:

$$C_{ph} = \frac{N_{slots}}{N_{ph}} = \frac{36}{3} = 12 \tag{7.15}$$

Number of sections:

$$S = HFC(C_{ph}, p) = HFC(12, 6) = 6 \tag{7.16}$$

Number of slots per section:

$$N_{SS} = \frac{N_{slots}}{S} = \frac{36}{6} = 6 \tag{7.17}$$

The number of slots forward to the next coil, S_F, is then obtained from:

$$S_F = N_{SS} - \sigma_{max} = 6 - 3 = 3 \tag{7.18}$$

From that, it is known that the second coil will start in the first coil's end-slot plus 3 i.e. $4 + 3 = 7$. Since the polarity has to be alternated for each coil, the end slot will be $7 - 3 = 4$. Where the -3 is $-\sigma_{max}$. This coil can be seen in figure 7.3. Note that the polarity arrow is in opposite direction compared to the first coil. Before continuing with more coils, it is best to calculate the number of coils in one sequence:

$$C_{seq} = \frac{C_{ph}}{S} = \frac{12}{6} = 2 \tag{7.19}$$

Since the number of coils per sequence is 2, the first sequence is done. The subsequent sequences are obtained by copying the previous sequence $N_{SS} = 6$ slots forward. The second sequence's first coil, therefore starts in the first sequence first coils start-slot plus N_{SS}, i.e. $1+6 = 7$. The second sequence is identified by double arrows on the coils in figure 7.3. Subsequent sequences are identified with a number of arrows equal to the sequence number.

The winding for phase two and three equals the first one but with an offset to the previous by:

$$Offset = \frac{2}{3}\frac{N_{slots}}{2p} + k\frac{N_{slots}}{p} = \frac{2 \cdot 36}{3 \cdot 2 \cdot 6} = 2 \tag{7.20}$$

where k in equation 7.12 is 0. The first coil in phase 1 started in slot 1 and ended in slot 4. With the above offset, that means that the first coil in phase 2 will start in slot $1+offset = 1+2 = 3$ and end in slot $4+offset = 4+2 = 6$. The third phase coils is obtained by adding the offset to the second phase coils. Since the first coil in phase two starts in slot 3 and ends in slot 6, the first coil in phase three will start in slot $3 + offset = 3 + 2 = 5$ and end in slot $6 + offset = 6 + 2 = 8$.

The complete winding for phase one is shown in figure 7.3 but the two others are omitted for clarity.

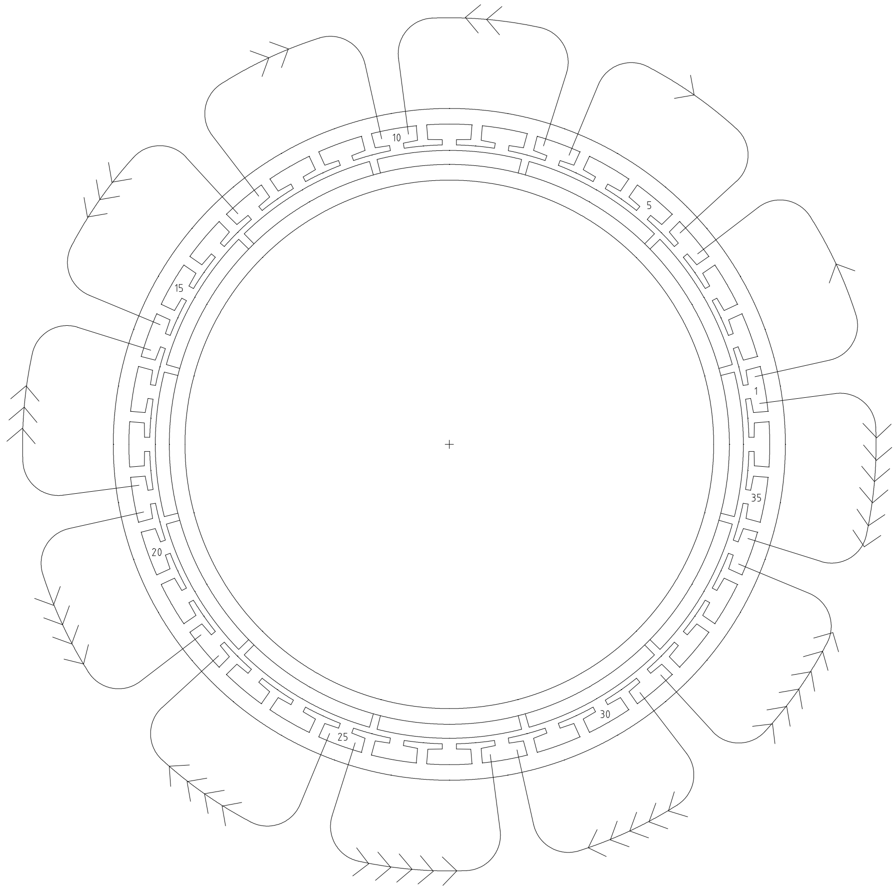

Figure 7.3: Winding for 36 slots, 12 poles, arrow direction shows polarity, number of arrows equals sequence

7.2.2 Winding for 12 poles, 27 slots

The windings for a 12 poles, 27 slots configuration is obtained like previously described. First the maximum coil span:

$$\sigma_{max} = NLI\left(\frac{N_{slots}}{2p}\right) = NLI\left(\frac{27}{2 \cdot 6}\right) = 2 \tag{7.21}$$

With a maximum coil span of two, the first coil starts in slot 1 and ends in slot $1 + 2 = 3$. This first coil can be seen in figure 7.4. As before slot one is to the right and the first one totally above the center of the motor.

$$\frac{N_{slots}}{2p} = \frac{27}{12} = \sigma_{max} + \epsilon = 2 + 0.25 \tag{7.22}$$

In this case ϵ becomes 0.25. It is nonzero since this is a fractional slot design. But since it is still less then 0.5 the first case in equation 7.6 will still be used. Some variables to be calculated before the slot forward parameter can be calculated: Number of coils per phase:

$$C_{ph} = \frac{N_{slots}}{N_{ph}} = \frac{27}{3} = 9 \tag{7.23}$$

Number of sections:

$$S = HFC(C_{ph}, p) = HFC(9, 6) = 3 \tag{7.24}$$

Number of slots per section:

$$N_{SS} = \frac{N_{slots}}{S} = \frac{27}{3} = 9 \tag{7.25}$$

And now the number of slots forward to next coil is given by:

$$S_F = N_{SS} - \sigma_{max} = 9 - 2 = 7 \tag{7.26}$$

A S_F of 7 means that the second coil will start in slot $3 + 7 = 10$ where 3 is the first coil's end slot. The end slot will be $10-2 = 8$ as the second coil have opposite polarity. The third coil will start in the second coil's end slot plus S_F i.e. $8+7 = 15$ and end in the start slot plus σ_{max}, i.e. $15+\sigma_{max} = 15+2$.

The number of coils in each sequence are:

$$C_{seq} = \frac{C_{ph}}{S} = \frac{9}{3} = 3 \tag{7.27}$$

So there are three sequences each displaced $N_{SS} = 9$ slots from each other. The whole winding for phase one including all tree sequences are displayed in figure 7.4. The number of arrows on the coils in the figure indicates the sequence number.

The offset for the other phases are:

$$Offset = \frac{2}{3}\frac{N_{slots}}{2p} + k\frac{N_{slots}}{p} = \frac{2 \cdot 27}{3 \cdot 2 \cdot 6} + 1 \cdot \frac{27}{6} = 6 \tag{7.28}$$

where the lowest k to get a integer offset is 1.

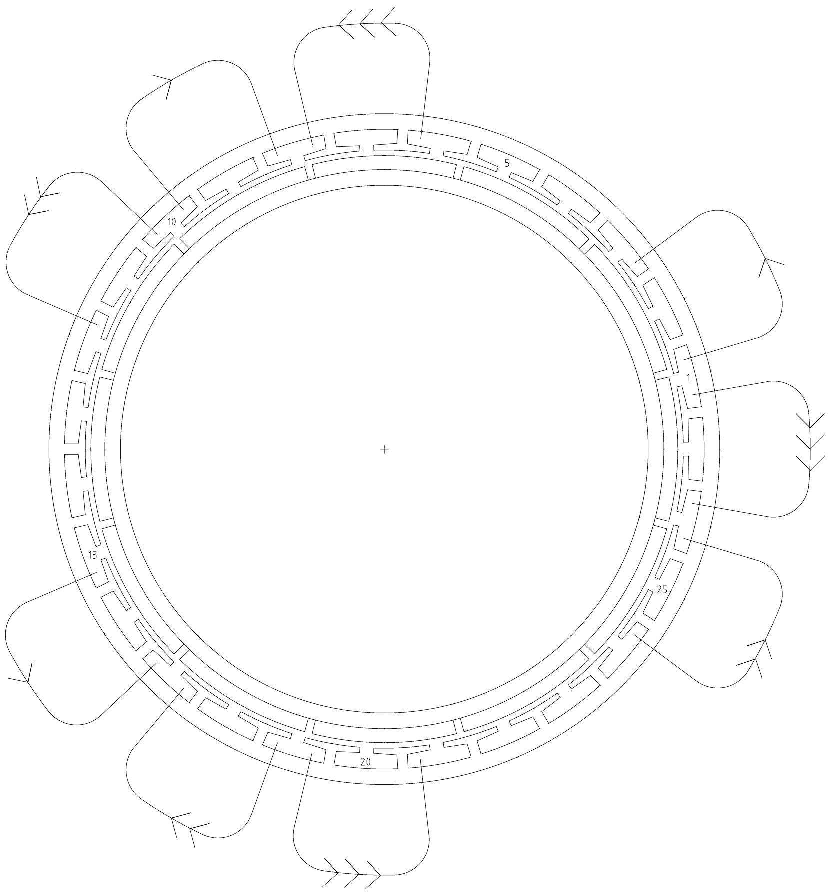

Figure 7.4: Winding for 27 slots, 12 poles, arrow direction shows polarity, number of arrows equals sequence

7.2.3 Comparison of 36 slots and 27 slots

Comparison of the 36 slots design to the 27 slots design is done in section 5.3.

Chapter 8

Calculations

The behavior of an electric machine depends of the distribution of the magnetic field. To know how the field is distributed around the motor, methods like the finite element method can be used. After a finite element method calculation, the field is known to magnitude and direction everywhere in the motor. The drawback with the finite element method is that it is difficult to know how different parameters depends of each other. For example how a change of the thickness of one part of the motor affects the thickness of another part. Because of that, analytical calculations is used in the design process to get a better feeling for how the machines behavior depends upon different design parameters. A common method is the magnetic circuit analysis, where the direction of the field is assumed to be known. From that it is possible to develop a model of the same kind that is used in electric circuit analysis.

The magnetic field is described by two quantities. Flux density B and field intensity H. To have a feeling for these quantities simplifies the understanding of electric machines.

- The flux density B, is the amount of magnetic field flow per area through a material. The unit for B is Tesla, T
- The field intensity H, is the resulting changes in the intensity of the magnetic field due to the interaction of B with the material it goes through. The unit for H is Amperes per meter.

The relation between B and H in linear materials is $B = \mu H$. Where μ is a constant called permeability. This relation is usually used as an approximation for materials in electric motors even if the true relation is a lot more complicated.

8.1 Magnetic circuit formulation

The magnetic circuit analysis reminds of the use of Ohm's law in electric circuits. To obtain the basic magnetic circuit equation two quantities have to be introduced. One is the total flux through a surface. This total flux ϕ is obtained in an intuitive way by multiply the flux density B, with the total area A:

$$\phi = BA \tag{8.1}$$

This is like obtaining the total electric current through a material by multiplying the current density with the area. The other quantity, magnetomotive force (mmf), $\mathscr{F}$ is obtained in a similar manner. It's obtained by multiply the field intensity H with the length l of the material, (in the direction perpendicular to the area):

$$\mathscr{F} = Hl \tag{8.2}$$

This corresponds to obtaining the voltage drop over a material in the electric circuit theory by multiplying the electric field (V/m) by the path length.

The first of these two equations is solved for B and the second one for H. The expressions for B and H are put in to the expression:

$$B = \mu H \tag{8.3}$$

After some rearrangement the fundamental equation for magnetic circuits is obtained:

$$\begin{gathered} \phi = P\mathscr{F} \\ [Wb] = [Wb/A][A] \end{gathered} \tag{8.4}$$

where

$$P = \frac{\mu A}{l} \tag{8.5}$$

P is called the permeance of the material identified by the area A, length l and permeability μ. The flux ϕ grows as P grows for a given $\mathscr{F}$. This can be compared to ohm's law:

$$I = GV = \frac{1}{R}V \tag{8.6}$$

(I current, V voltage, R resistance, G conductance) Where the current, I grows as the conductance, G grows (or R decrease) for a given voltage V. It's however an important difference between the magnetic circuit law and Ohm's law. Ohm's law gives how much power that will be dissipated in the material. In the magnetic circuit equivalent no power will be dissipated instead the power will be stored magnetically in the material. A more detail study of P gives: The flux ϕ for a given magnetomotive force $\mathscr{F}$ grows with μ and A and decrease for an increase of l. The inverse of P is called reluctance:

$$\mathscr{R} \equiv \frac{1}{P} \tag{8.7}$$

The basic theory of magnetic circuits is now introduced. In the subsequent text, magnetic circuit models for the different parts of a brushless permanent motor will be explained.

8.2 Circuit models

In this section magnetic circuit models for the parts in the motor will be presented [9, page 34].

8.2.1 Coil circuit model

A coil is represented in a magnetic circuit with a field source (compare with a voltage source in an electric circuit), see figure 8.1. The magnetomotive force for the field source is given by:

$$\mathscr{F}_c = NI \tag{8.8}$$

where I is the current through the coil and N the number of turns. In addition to this, the coils core is represented as a permeance in series with the field source according to:

$$P_c = \frac{\mu A}{l} \tag{8.9}$$

That permeance can also be seen in the figure.

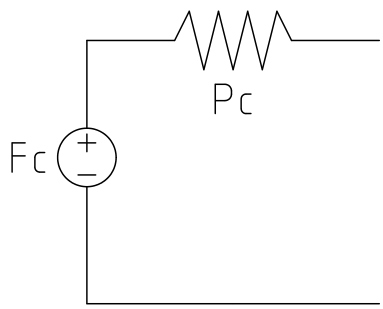

Figure 8.1: Magnetic circuit model for a coil with core. (Fc = $\mathscr{F}_c$, Pc = P_c)

8.2.2 Magnetic circuit model for permanent magnet

As mentioned before the magnet is working on a line between $(0, B_r)$ and $(-\mu_0 H_c, 0)$. This line can be described by:

$$B_m = B_r + \mu_R \mu_0 H_m \tag{8.10}$$

Where B_m and H_m are the current magnetic flux intensity and magnetic field which depends on the permeance between the magnet poles. This equation can be rewritten in a form like the basic magnetic circuit equation. By multiplying with the magnets cross-sectional area, A_m, we obtain:

$$\phi_m = B_r A_m + \mu_R \mu_0 A_m H_m \tag{8.11}$$

The first term on the right hand side is constant and defined as ϕ_r. The second term is rewritten with use of equation 8.2 ($\mathscr{F} = Hl$) and the definition permeance of, 8.5, ($P = \mu A/l$). The equation becomes:

$$\phi_m = \phi_r + P_m \mathscr{F}_m \tag{8.12}$$

where

$$P_m = \frac{\mu_R \mu_0 A_m}{l_m} \tag{8.13}$$

A permanent magnet can be seen as a flux source in parallel with a permeance. See figure 8.2. If no flux is leaving the magnet all ϕ_r goes through the permeance P_m which results in a magnetomotive force $\mathscr{F}_m$. This is the open circuit magnetomotive force.

Leakage flux

The permanent magnets circuit model have now been discussed. Another issue, important when magnets are mounted on a rotor, will also be discussed in this section about permanent magnet circuit models. In a brushless motor the magnet is in principle mounted as in figure 8.3. In the figure the main flux path is illustrated by the outer closed line with arrows. The flux path starts in one magnet. From where is goes over the air gap into the stator. In the stator it pass through one coil into the stator back iron and through the stator back iron, through another coil and out in the air gap. From the air gap it goes into another magnet and then through the rotor back iron back to the first magnet. All this flux is useful since it goes through the coils.

There is however another flux path that the flux can take. That one is illustrated with the inner line with arrows. The flux that goes out of the first magnet close to the other magnet can instead of going into the stator, go

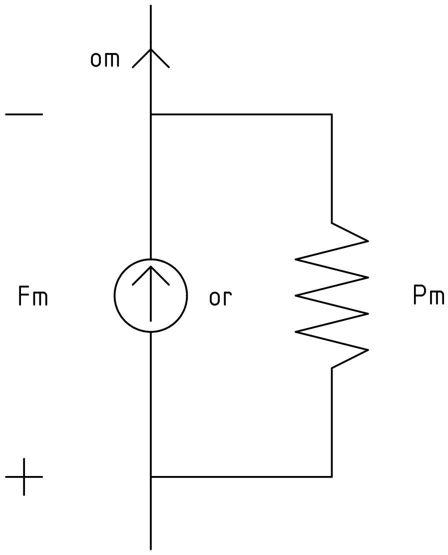

Figure 8.2: Magnetic circuit model for a permanent magnet. (om = ϕ_m, Fm = $\mathscr{F}_m$, or = ϕ_r, Pm = P_m)

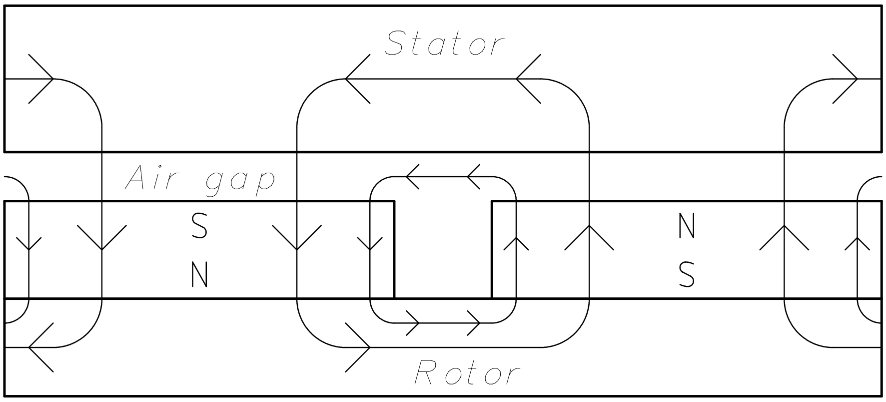

Figure 8.3: Main flux path and flux leakage path.

directly to the other magnet. This flux doesn't go through the coils in the stator and is therefore of no use. It is called leakage flux. How big this flux will be depends on the permeance of the path for the flux linkage in the air gap, since the air part of the path is the most limiting for the flux.

The permeance of this path can be calculated by calculating the sum of infinitely small pieces of the path. According to figure 8.4, each of these small pieces have an area of Ldx, there L is the length of the motor in axial direction. The length of one piece is, first arc $\pi x/2$, the straight part τ_f and the last arc $\pi x/2$. In total $\tau_f \pi x$. The permeance can be obtained by using the ordinary permeance formula and integrate over all these small pieces:

$$\begin{aligned} P_{ml} &= \frac{\mu_0 A}{l} = \int_0^g \frac{\mu_0(Ldx)}{\tau_f + x\pi} = \frac{\tau_f \mu_0 L}{\pi \tau_f}\left[\ln(1+\frac{x\pi}{\tau_f})\right]_0^g \\ &= \frac{\mu_0 L}{\pi}\ln\left(1+\frac{g\pi}{\tau_f}\right) \end{aligned} \tag{8.14}$$

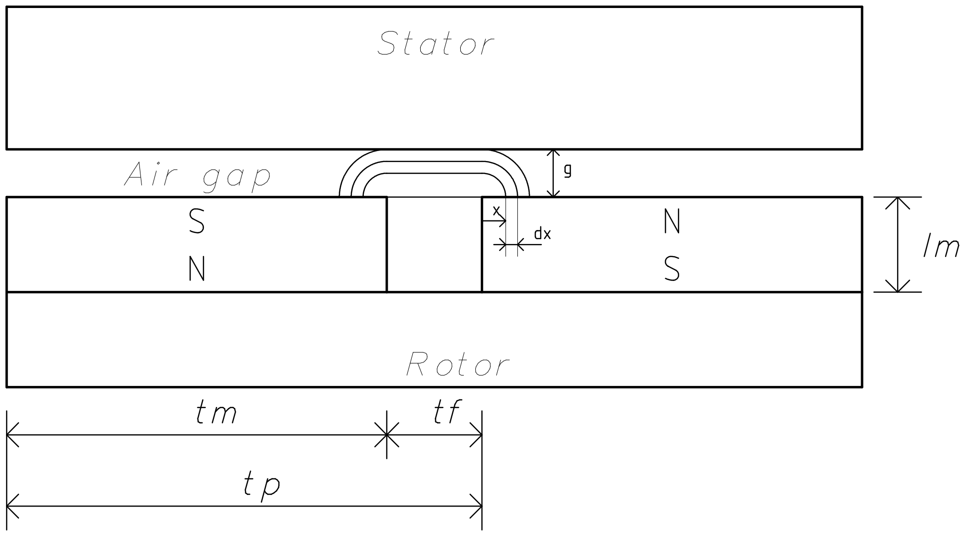

Figure 8.4: Dimension used to calculate the leakage permeance. tm=τ_m, tf = τ_f and tp = τ_p

8.2.3 Air gap circuit model

In this design there are no air gaps, only water gaps. However the magnetic properties of water is very close to that of air, so the water gap will be treated as an ordinary air gap. An air gap is modeled as a pure permeance (compare with a resistance in an electric circuit). A straightforward approach to model the air gap in a motor with slots, is to model the airspace as a rectangular parallelepiped (box) and use the ordinary permeance formula $P = \mu A/l$. This gives however not an accurate value since the flux don't go only in a straight line between the iron cores. The path the field takes is illustrated in figure 8.5. Most of the field goes straight over the air gap but there are also fringing flux that goes beside the main path. This flux goes from the upper material in the figure and in to the tooth a bit down as indicated by the arc shaped line. To calculate the permeance for the flux on one side of the tooth,

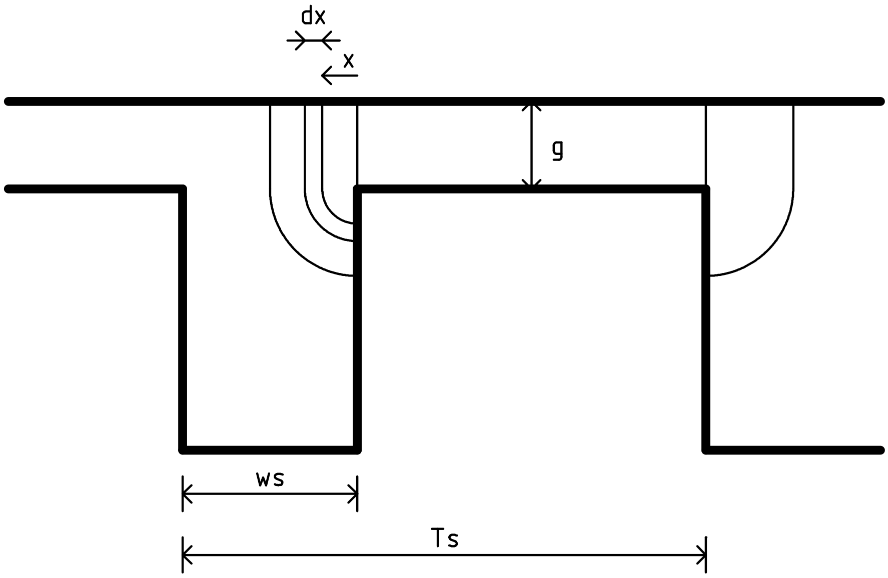

Figure 8.5: Magnetic circuit model for an air gap with slots. (ws=ω_s, Ts = τ_s)

the arc shaped area can be divided in infinitesimal small piece, as indicated by the dx in the figure and use the ordinary formula $P = \mu A/l$. The area for each infinitesimal piece equals the length L of the motor times dx. The length of the arc is given by $x\pi/2$ and the length of the straight part by g. The total length of the infinitesimal piece is given by the sum of these, i.e.: $x\pi/2 + g$. The area and total length of the infinitesimal piece is put into the formula for P and integrated over x to give the total P for the leakage flux for one side of the tooth,(See figure 8.5):

$$P_{oneside} = \frac{\mu_0 A}{l} = \int_0^{\frac{\omega_s}{2}} \frac{\mu_0 L dx}{g + \frac{x\pi}{2}} \tag{8.15}$$

With some math the following is obtained:

$$P_{oneside} = \frac{\mu_0 L}{g} \int_0^{\frac{\omega_s}{2}} \frac{1}{1 + \frac{\pi x}{2g}} dx = \frac{2\mu_0 L}{\pi} \ln(1 + \frac{\pi\omega_s}{4g}) \tag{8.16}$$

The permeance due to the field that goes straight between the two top and bottom of the figure can easily be calculated by:

$$P_{straight} = \frac{\mu_0 L(\tau_s - \omega_s)}{g} \tag{8.17}$$

where L is the length of the motor, τ_s the pole-pitch and ω_s the width of the tooth. The two last variables can be seen in the figure.

The total air gap permeance can be calculated by adding two times the permeance for the fringing flux permeance on the side to the permeance for the straight part. The total permeance then becomes:

$$P_g = P_{straight} + 2P_{oneside} = \mu_0 L[\frac{\tau_s - \omega_s}{g} + \frac{4}{\pi}\ln(1 + \frac{\pi\omega_s}{4g})] \tag{8.18}$$

It is convenient to treat the slotted air gap as an uniform air gap. This can be done by introducing an effective air gap scaling factor, called Carter coefficient. This mean that the slots are removed in the math model and the stator is seen like a flat surface. The removed slots is compensated for by a bigger air gap. The above expression for P_g (equ 8.18) can be used to obtain that type of scaling factor [9, page 97]:

$$k_c = \frac{1}{1 - \frac{\omega_s}{\tau_s} + \frac{4g}{\pi\tau_s}\ln(1 + \frac{\pi\omega_s}{4g})} \tag{8.19}$$

With use of this coefficient the air gap permeance can easily be calculated by:

$$P_g = \frac{\mu_0 A}{g k_c} \tag{8.20}$$

8.2.4 Magnetic circuit model for a brushless motor

The magnetic circuit model is plotted in figure 8.6. The corresponding flux to the model in the figure is the main flux loop in figure 8.3. P_s and P_r are the stator and rotor permeance respectively and they are both very high and can therefore be ignored. P_m is the permeance for the magnet as mentioned in the section for the circuit model for permanent magnet. It's calculated by applying the ordinary formula for the permeance of a rectangular parallelepiped (box):

$$P_m = \frac{\mu_R \mu_0 \alpha_m \tau_p L}{l_m} \tag{8.21}$$

See figure 8.4 for definitions of the variables. α_m is defined as τ_m/τ_p. L is the length of the motor in axial direction. In the circuit model $P_m/2$ is used, this because only half of the magnet is used in one flux path, (compare to an electric circuit where the conductance is smaller if you only use half of a conductor).

The flux leakage permeance P_{ml} is calculated as described in section 8.2.2. τ_f in the formula in that section can be expressed in terms of the magnet

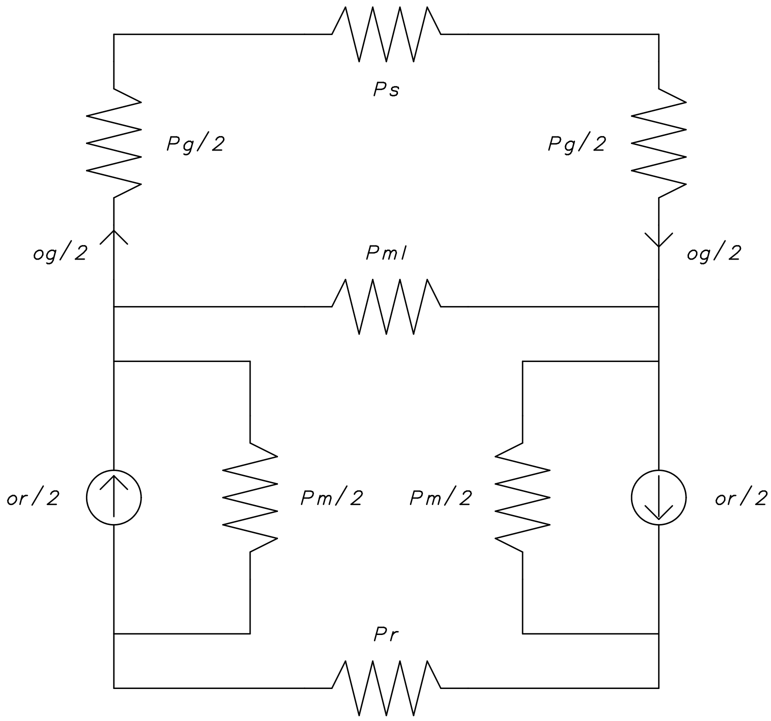

Figure 8.6: Magnetic circuit model for a brushless motor. (or = ϕ_r, og = ϕ_g)

width τ_p, as $(1-\alpha_m)\tau_p$. P_{ml} then becomes:

$$P_{ml} = \frac{\mu_0 L}{\pi} \ln\left(1 + \frac{g\pi}{(1-\alpha_m)\tau_p}\right) \tag{8.22}$$

The only permeance in the figure that have not been discussed is the air gap permeance, P_g. When a magnetic model for the radial motor design is developed, the stator is modeled without slots. The slots comes in to the equation implicitly through the effective air gap coefficient described before. The stator can therefore be treated as one big area ($\tau_p L$). The ordinary formula for permeance of a box can't be directly applied to the air gap even with the stator as a homogeneous surface. This because the area of the magnets facing the air gap ($\tau_p \alpha_m L$) is smaller than the stator area ($\tau_p L$). An approximate value can be obtain using the mean of these two areas:

$$A_g = L\frac{\tau_p + \tau_p\alpha_m}{2} = \frac{\tau_p L(1+\alpha_m)}{2} \tag{8.23}$$

The air gap permeance then becomes:

$$P_g = \frac{\mu_0 \tau_p (1+\alpha_m) L}{2g_e} \tag{8.24}$$

Where g_e equals gk_c where k_c is the Carter coefficient from equation 8.19. In equation 8.19 the side of the air gap opposite to the slotted surface was assumed to be an iron core. In this case the side opposite to the slotted stator consist of the magnets. In that case the actual air gap length g can't be used directly to calculate k_c since a magnet doesn't behave like an iron core. It have however been shown [10] that the magnets and the actual air gap can be combined to an effective air gap when calculating the Carter coefficient. The air gap length to be used when calculating the Carter coefficient is therefore:

$$g_c = g + \frac{l_m}{\mu_R} \tag{8.25}$$

Where l_m is the length of the magnets in radial direction as shown in figure 8.4. This means that for an ordinary permanent magnet with μ_R around 1 the magnet is treated like air and the air gap is considered to go all the way from the stator to the rotor back iron. (Compare this with electric steel with a μ_R on several thousands which will make the second term disappear and leave $g_c = g$)

In the circuit model $P_g/2$ is used as only half of the air gap is used in one flux path.

All the permeances in the circuit model for the motor have now been studied. Next step is to simplify the model a bit. The two models that each represent half of a magnet($P_m/2$ combined with $\phi_r/2$), can be replaced by one model representing the two in series. The two air gap permenaces can in the same way be combined to one. Keep in mind that the P_s and P_r already have been neglected. The circuit model then looks like in figure 8.7. The equation for the flux $\phi_g/2$ in the air gap loop can when be written as:

$$\frac{\phi}{2} = \frac{P_g/4}{P_g/4 + \widetilde{P}_m/4}\frac{\phi_r}{2} \tag{8.26}$$

There $\widetilde{P}_m = P_m + 4P_{ml}$ is $P_m/4$ in parallel with P_{ml}. $\widetilde{P}_m$ can be seen as an effective magnet permeance where the magnet leakage have been included in the model for the magnet. It's usually convenient to express $\widetilde{P}_m$ as P_m times a factor. This can be done by using equation 8.21 for P_m and equation 8.22

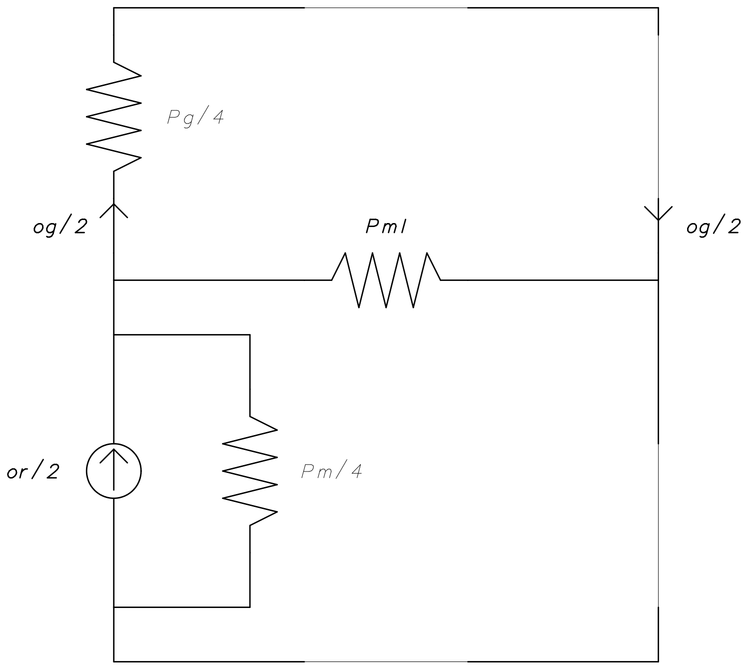

Figure 8.7: Simplified magnetic circuit model for a brushless motor. (or = ϕ_r, og = ϕ_g)

for P_{ml}:

$$\begin{aligned}\widetilde{P}_m &= P_m + 4P_{ml} \\ &= \frac{\mu_R\mu_0\alpha_m\tau_p L}{l_m} + 4\frac{\mu_0 L}{\pi}\ln\Big(1+\frac{g\pi}{(1-\alpha_m)\tau_p}\Big) \\ &= \frac{\mu_R\mu_0\alpha_m\tau_p L}{l_m}\Big(1+4\frac{l_m}{\pi\mu_R\alpha_m\tau_p}\ln\Big(1+\frac{g\pi}{(1-\alpha_m)\tau_p}\Big)\Big) \\ &\equiv P_m k_{ml}\end{aligned} \tag{8.27}$$

The total air gap flux is then:

$$\phi_g = \frac{1}{1+\widetilde{P}_m/P_g}\phi_r \tag{8.28}$$

The circuit model for the motor is now simplified as much as possible. Next step is to insert the previously obtained variables in the above equation. P_g in the above equation is replaced by the expression in equation 8.24, with $g_e = gk_c$. The last expression in equation 8.27 is inserted for $\widetilde{P}_m$ with P_m

replaced by equation 8.21. We then obtain:

$$\phi_g = \frac{1}{1 + \frac{2\mu_R \alpha_m k_{ml} k_c g}{(1+\alpha_m) l_m}} \phi_r \tag{8.29}$$

The interpretation of this expression can be simplified by some algebraic work. A first tricky thing is to see that the flux concentration factor C_ϕ is hiding in the denominator. The flux concentration factor is defined by equation 6.2 and is given by:

$$C_\phi = \frac{A_m}{A_g} = \frac{\alpha_m \tau_p L}{\tau_p L(1+\alpha_m)/2} = \frac{2\alpha_m}{1+\alpha_m} \tag{8.30}$$

where A_g comes from equation 8.23. Compare this expression with the denominator in the expression for ϕ_g and you get:

$$\phi_g = \frac{1}{1 + \frac{\mu_R k_{ml} k_c g C_\phi}{l_m}} \phi_r \tag{8.31}$$

The expression can be simplified even further by comparing the current expression with equation 6.1, ($P_c = l_m/(gC_\phi)$). This gives:

$$\phi_g = \frac{1}{1 + \frac{\mu_R k_{ml} k_c}{P_c}} \phi_r \tag{8.32}$$

Since B_r is given by the magnet manufacturer, it's convenient to use that instead of ϕ_r. This gives:

$$B_g = \frac{C_\phi}{1 + \frac{\mu_R k_{ml} k_c}{P_c}} B_r \tag{8.33}$$

It's desirable to obtain as big flux density B_g as possible for a given magnets B_r value. Studying the above equation then gives that C_ϕ should be as big as possible. From equation 6.2 we have, $C_\phi = 2\alpha_m/(1+\alpha_m)$ implying that α_m should be as big as possible, (i.e.1), meaning, not surprisingly that the magnets should cover as big portion of the air gap as possible. However $C_\phi = l_m/(P_c g)$, meaning that an increase must imply longer magnets and that is very undesired in this application, or smaller g, or smaller P_c. g should be kept as small as possible without getting the rotor touching the stator at any circumstances. P_c should be small, but in motor design a value below 4 is undesired.

Next study the denominator that obviously should be as small as possible. μ_R is fixed by the magnet material around 1 and can't be affected. k_c given by equation 8.19, don't vary much with affectible variables, the affectible part of the denominator is very small compared to the 1 in the denominator. Studying equation 8.27 gives that k_{ml} is minimized when g/τ_p is minimized. We always want g to be as small as possible. τ_p is highly influenced by basic geometrical variables and the number of magnets poles that may be more important than reducing k_{ml}. Another way to minimize k_{ml} is to minimize $l_m/(\alpha_m \tau_p)$. Usually $l_m/(\alpha_m \tau_p)$ is kept below 1/4 [9, page 68]. α_m is usually given by other constraints.

The last variable in the denominator is P_c. It should be chosen as big as possible to maximize B_g for a given B_r. A high P_c is good in general for a good magnetic performance but conflict with the need for a small C_ϕ in the numerator.

To summaries, there are a lot of trade-offs to get a big B_g for a given B_r. To be able to find a good optimization, a change of the variables in equation 8.33 and the variables they depend on, must be weighted with how they affect the total expression in the current setup.

8.3 Torque generation equations

When a whole coil with n_s turns is aligned to a magnet that gives a flux of ϕ_{gm} in the air gap, a flux linkage of $n_s \phi_{gm}$ is obtained. When the rotor rotates to the next magnet that have opposite polarity, the same flux linkage is obtained but with opposite sign. The flux linkage then increases from $-n_s \phi_{gm}$ to $+n_s \phi_{gm}$ from the middle of that magnet to the middle of the next. This creates a triangle wave. The distance between a maximum and a minimum of this wave equals the space between two magnets centers. It is given by:

$$\tau_p = \frac{2R_{ro}\pi}{N_m} \tag{8.34}$$

where R_{ro} is outer rotor radius, 2 to get the diameter, π to get the circumference and N_m the number of magnets. Since the back-EMF e is proportional to the derivative of the flux-linkage according to equation 7.3, the back-EMF must be a square-wave. When the flux linkage increases (at a constant rate), the back-EMF have a constant positive value equal to the slope of the flux-linkage times ω_m. When the flux-linkage decrease the back-EMF is the same, but with opposite sign because of the negative slope of the flux-linkage wave. Equation 7.3 express the change of flux linkage over a change in angle rather than in change of distance. The angle between a maximum and a minimum of the flux linkage is:

$$\theta_p = \frac{2\pi}{N_m} \tag{8.35}$$

The magnitude of the back-EMF when all coils for one phase (N_{cp}) are connected in series becomes:

$$|e| = N_{cp}\frac{n_s\phi_{gm} - (-n_s\phi_{gm})}{\theta_p}\omega_m = N_{cp}\frac{2n_s\phi_{gm}}{\theta_p}\omega_m \tag{8.36}$$

It is more convenient to express the flux form one magnet, ϕ_{gm} in terms of the flux density B_g. The flux equals:

$$\phi_{gm} = \tau_p\alpha_m LB_g = \theta_p R_{ro}\alpha_m LB_g \tag{8.37}$$

where α_m is the fraction of the rotor circumference covered by magnets. Expression 8.36 then becomes:

$$|e| = N_{cp}2n_s R_{ro}\alpha_m LB_g\omega_m \tag{8.38}$$

Where B_g is given by equation 8.33. Since two conductors are conducting at every time instance in a three phase motor the line to line voltage magnitude becomes $2|e|$ for a wye connected machine and $|e|$ for a delta connected.

When the back-EMF is known it's possible to calculate the torque from equation 7.4:

$$T = \frac{2|e|i}{\omega_m} = 2\cdot N_{cp}2n_s R_{ro}\alpha_m LB_g i \tag{8.39}$$

The 2 is there because a wye connection is assumed. The current i is the current through the two phases in series currently conducting. That means i is also the line current.

Usually $|e|$ and T is given and we want to calculate the number of turns n_s and the current i to obtain the specified $|e|$ and T. n_s can be obtained by rearranging equation 8.38:

$$n_s = \frac{|e|}{N_{cp}2R_{ro}\alpha_m LB_g\omega_m} \tag{8.40}$$

This value have to be rounded off to a integer value. The current is then given by rearranging equation 8.39 to:

$$i = \frac{T}{2\cdot N_{cp}2n_s R_{ro}\alpha_m LB_g} \tag{8.41}$$

8.3.1 Torque dependence of coil span

In the previous section the coil span, i.e. the length of the coil in the direction of motor circumference, was equal to the length between the centers of two adjacent magnets. In figure 8.8 the coil span has been decreased to 0.75 of a magnet length. In this case the maximum flux linkage becomes smaller

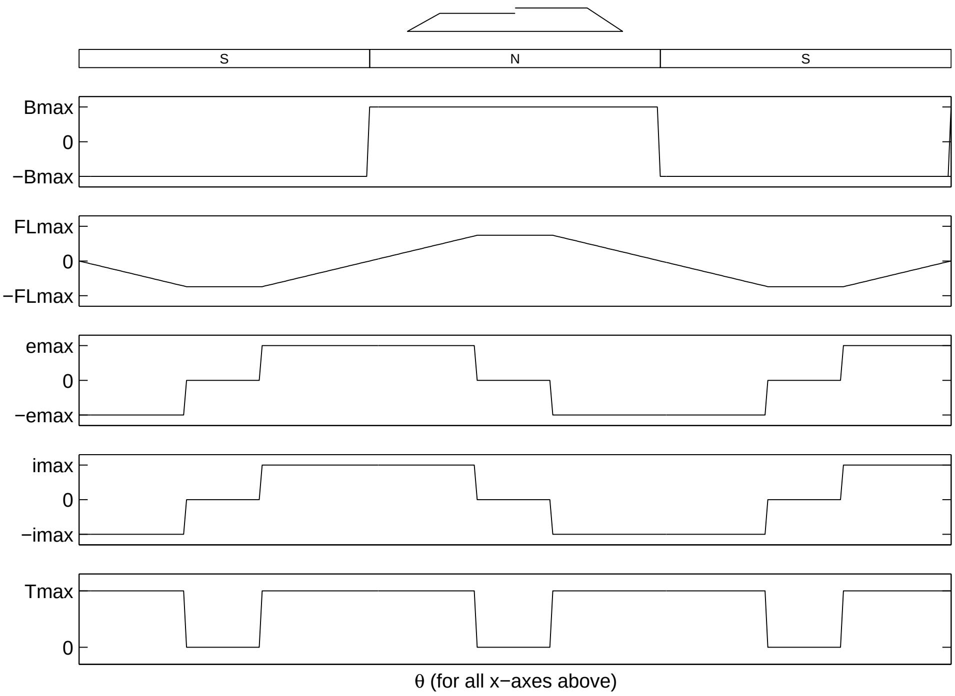

Figure 8.8: Waveforms from basic rectangular coil/magnet interaction when the coil at the top moves from one side to the other. More magnets follows to the left and right of the figure. Coil span 0.75 of magnet width. (FL = Ψ)

according to equation 7.1 since the area of the coil is smaller. The interesting aspect is however to study the change of the flux linkage as the coil moves relative to the rotor magnets, since that is what gives the back-EMF. When the coil cross over from one pole to another the rate of change of the flux linkage will be the same as in the previous example. To understand that, think about two points in time. At the first point the coil's middle is just between a N-pole and a S-pole magnet. At the second point the coil have moved ϕ in direction towards the S-pole magnet. This means that the coil area over the N-pole magnet have decreased by $d\phi L$, where L is the length of the motor in axial direction. Axial direction corresponds to, into the paper in figure 8.8. The same applies for the coil part over the S-pole magnet, but with opposite sign, i.e. the coil area have increased by $d\phi L$. From equation 7.1 it can be seen that the change of flux linkage is directly proportional to the change in area. The increase over the N-pole, $d\phi L$ and decrease over the S-pole $d\phi L$ only depends of the speed and the axial length of the motor but it doesn't depends of the length of the coil. That is why the change of flux linkage is independent of the coil span.

When the coil is entirely over a S-pole magnet, maximum negative flux linkage is obtained. When the coil moves in over a N-pole magnet the flux linkage increases with the speed discussed above. In this case when the coil length is less than the magnet, this increase continues until the back of the coil reach the junction of the S-pole and N-pole magnet. From that position the coil can move in this case 0.25 magnet length until the front of the magnet reach the next S-pole. During that movement the flux through the coil will be constant. A constant flux linkage implies that the back-EMF and the produced torque will be zero.

Since a coil span less then a magnet length implies zero torque for some position of the coil, other coils in the motor have to be arranged in a way so the torque is constant at all times.

8.4 Computer aid calculations

To get the analytical calculations described above done, a Matlab program was written. The code for the Matlab program is in the appendix. This program calculates the dimensions of all parts of the motor depending on the in-parameters. The result is presented numerically and graphically. The graphical presentation is done by plotting how the motor looks. This graphical presentation makes it easy to quick get an overview, to see if the design looks reasonable. Otherwise some dimension can be unrealistic and parts may even cross over each other. Even the windings are plotted in this figure to get an overview of the hole motor design. The graphical output from the Matlab program can be seen in figure 8.9 and a zoomed in view in figure 8.10. In addition the program also generates geometrical data to use in the FEM program.

8.4.1 FEM calculation

For the finite element calculations (FEM) a program called iMOOSE from Dept. of Electrical Machines(IEM) at Aachen University have been used. It is an open source program for Linux, which name is short for "innovative

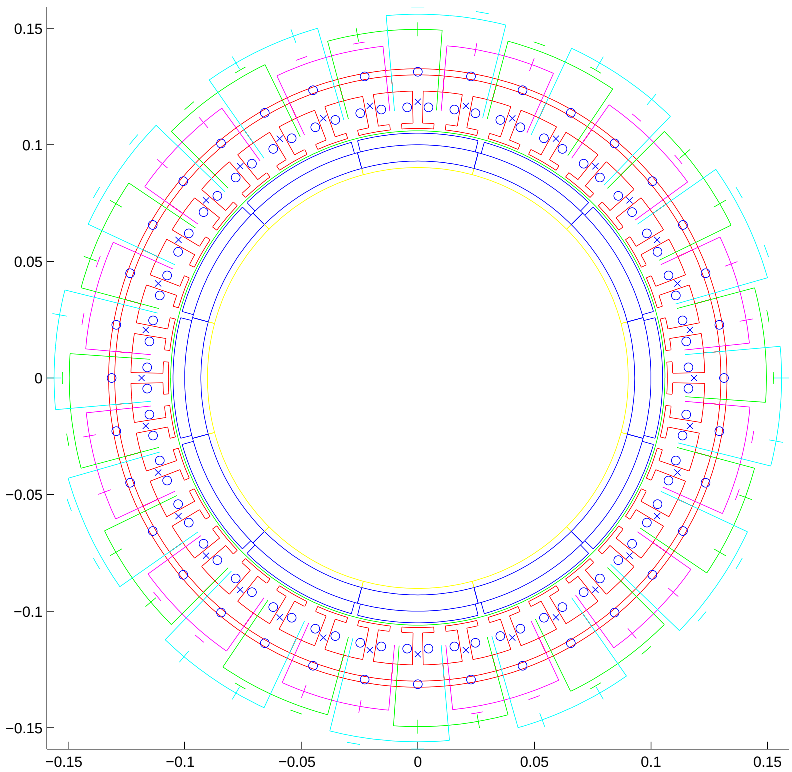

Figure 8.9: Output from Matlab program

Modern-Object Solving Environment.(It seems like the authors of the program also likes that the program name relates to the animal moose.) The program is actually a program package. There are a couple of programs in the package that can be used for different types of calculation setups. There are 2D and 3D programs and programs for time simulations and so on. When the FEM calculation is done the result can be graphically studied using the program trinity included in the package.

There are a couple of in data files for the calculation program. One set of files are the ones describing the meshed geometry of the motor, i.e. the motor layout subdivided in many small triangles for the FEM calculation. The main Matlab code described in the previous section also generates output to be used in the FEM calculations. The Matlab code generates a description of the motor geometry in form of an .poly file, suited for the mesh generation

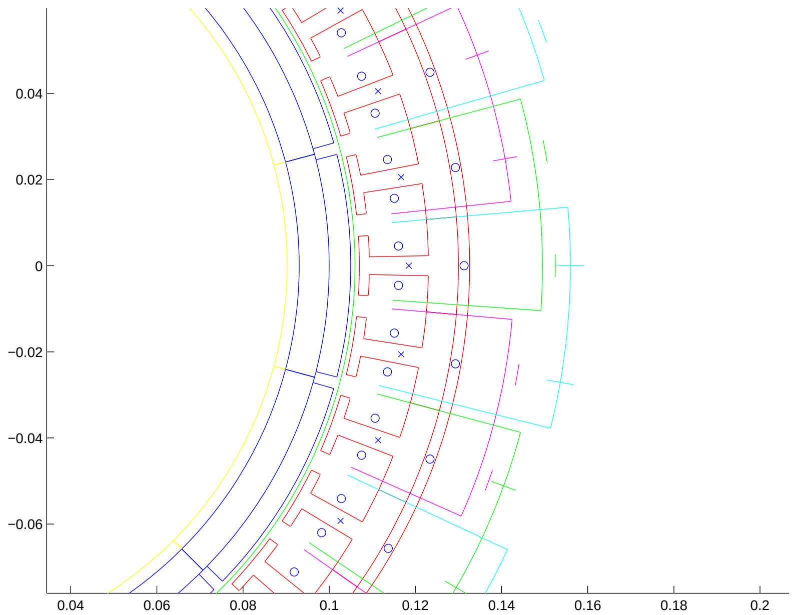

Figure 8.10: Output from Matlab program, zoomed

program. Parts of this file are listed here:

```
3558 2 0 0
1 1.060000E-01 0.000000E+00
2 1.059840E-01 1.850000E-03
3 1.059350E-01 3.699000E-03
...[many lines]...
3557 8.759300E-02 -2.678000E-02
3558 8.804700E-02 -2.524700E-02

6843 0
1 361 362
2 362 363
3 363 364
...[many lines]...
6841 318 317
6842 317 316
6843 316 3380

1
1  0  0

228
1 1.135000E-01 0.000000E+00 10 5.000000E-07
2 1.115582E-01 -4.346239E-03 371 5.000000E-07
3 1.115582E-01 4.346239E-03 300 5.000000E-07
...[many lines]...
```

```
226 8.876760E-02 -5.125000E-02 211 5.000000E-07
227 8.055083E-02 -4.650604E-02 100 5.000000E-07
228 9.136568E-02 -5.275000E-02 101 2.000000E-07
```

The .poly file consist of four parts. The first part is a list of all nodes (points) needed to describe the geometry of the motor.See the lines under 3558 in the above listing. All nodes are given a number, the left number above and coordinates, the number to the right above. The nodes are on the border of all parts of the motor. For flexibility the stator is divided in sections consisting of one shoe each and the rotor in sections consisting of one magnet each. The nodes are on the border of each of these sections and around the border of all different materials inside these sections. The sections can be studied in figure 8.12.

The different materials includes rotor and stator iron, rotor magnets, regions for conductors and air regions. For simplicity no boundary conditions have been used around the boundary of the hole design. Instead a layer of air have been placed both outside the stator and inside the rotor. These air regions makes the magnet field decrees to zero before the border of the calculation area is reached. In this approach the magnet field can be studied in the air and not only in the materials.

In the next section of the .poly file the created nodes are used to actually describe all material regions geometries. This is done by connecting the created nodes together. For example assume that there where four nodes created in the previous section each placed in a corner of a square and with number 1 to 4. In order to describe a square this section of the file states that node 1 should be connected to node 2, node 2 to node 3, node 3 to node 4, and finally node 4 to node 1. All parts of the motor is described in that manner. This connection of point are done under the number 6843 above, i.e. the first line tells that point 361 is connected to 362. The simple example above with points numbered 1..4 describes a square consisting of straight lines. In case of the motor all material regions have at least some sides that are arc shaped. To describe this, many nodes have to be placed on the border and connected together to get an good approximation of an arc shape. To make a region description make sense, all border descriptions have to form a closed region. It's not possible to have a region consisting of a square with only three sides even if there are another square next to it. It should however be noted that if one square is to be placed next to another only two new nodes should be created. The other corners of the square can use the same nodes as the neighboring square. If instead, new nodes are made it may be problems with the boundary conditions between the two squares. This fact makes the coding of the Matlab program that makes the .poly file a little bit tricky and it's even more tricky when the shared boundary is an arc, which is a common case.

Next section of the file is the hole section. In this section a logical hole point is placed in the origin of the rotor. If this is not done the hole area inside the rotor will be meshed. Since this is a very big area consisting of air it should not be meshed otherwise the number of triangles will increase very much and increase the calculation time a lot. In the above lusting there is one line with a 1 to tell that there is only one hole. After this line is a line starting with the hole number 1 and when two 0 to tell the position is origin i.e. (0,0).

The fourth and last part of the .poly file defines what material the different regions should have. For all regions defined in section two, this is done by specifying a coordinate within the regions and give this region a material number. This material number is then used in the main configuration file for the calculation program where all different material numbers are given material-properties defined in another set of files. In addition to specifying material numbers the last section can also be used to specify the maximum area of the FEM mesh triangles allowed in each region.

This .poly file serves as mentioned before as in data file to the mesh generation program. The mesh generation program used is Triangle. It's a freely available program written by Jonathan Richard Shewchuk at the Computer Science Division, University of California at Berkeley. The out data files is directly used by the calculation program.

In addition to the .poly-file describing the geometry of the motor some other data is generated by Matlab code. This other data goes to the main configuration file describing the problem setup, the .PROB-file. This includes the length of the motor, magnet and current excitation and a line describing how the calculation program should perform the torque calculation.

In addition to the data from the Matlab code there are some other settings that have to be set in the main configuration file.

Results from the FEM calculation

The numerical result of the FEM calculations was the torque. The torque was 20 Nm this is the desired value. In addition to the numerical result it is interesting to see the distribution of the magnetic flux density of the motor. The graphical program supplied with the FEM program packages to plot the result was a bit tricky to get running. Instead of using that, a program was written in the open source calculation program SciLab (www.scilab.org), to be able to view the magnetic flux density graphically. The magnetic flux density in in the whole design can be seen in figure 8.11. A more detailed version around one magnet is presented in figure 8.12. The direction of the flux can be seen in figure 8.13.

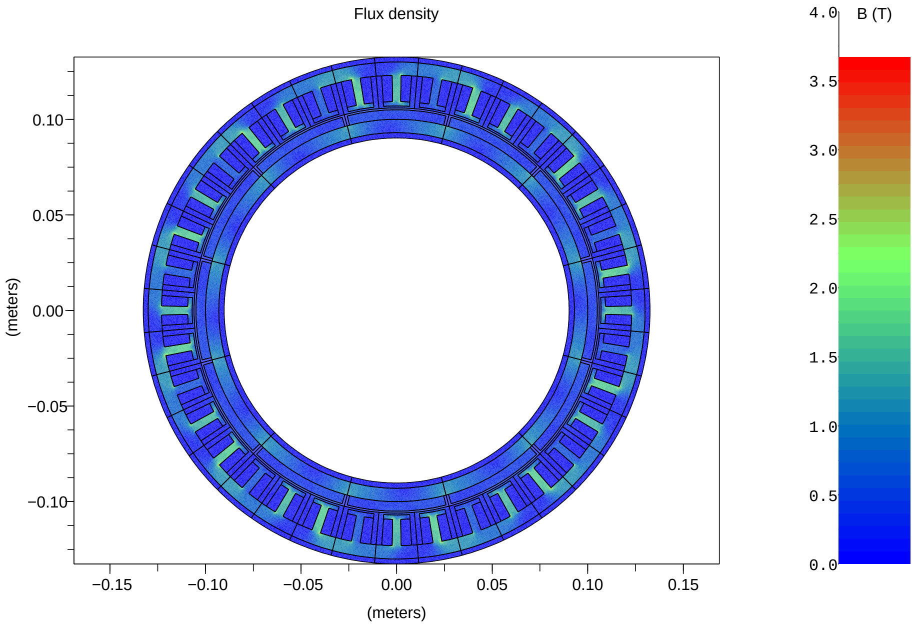

Figure 8.11: Flux density from FEM calculations

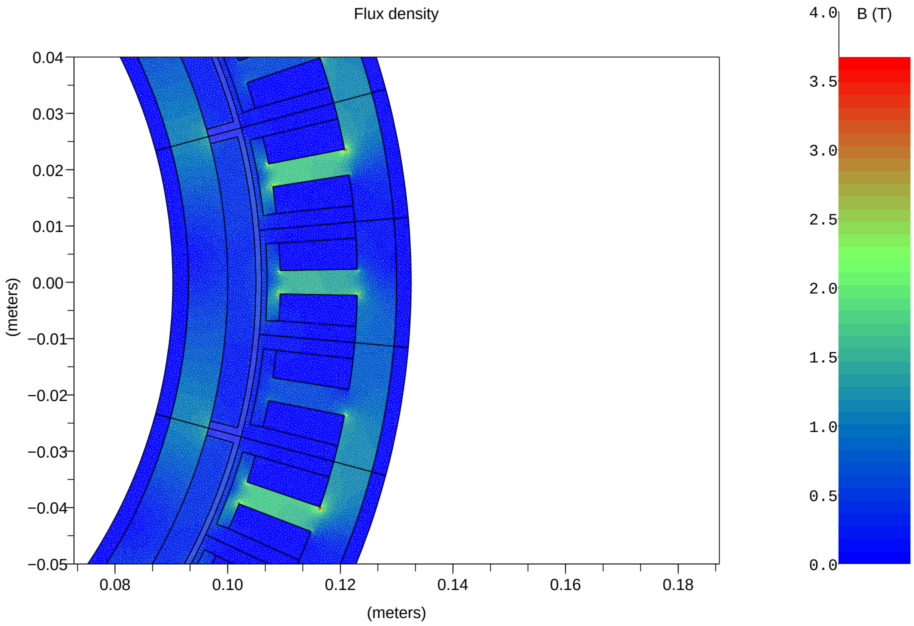

Figure 8.12: Flux density from FEM calculations. Zoomed to a magnet section.

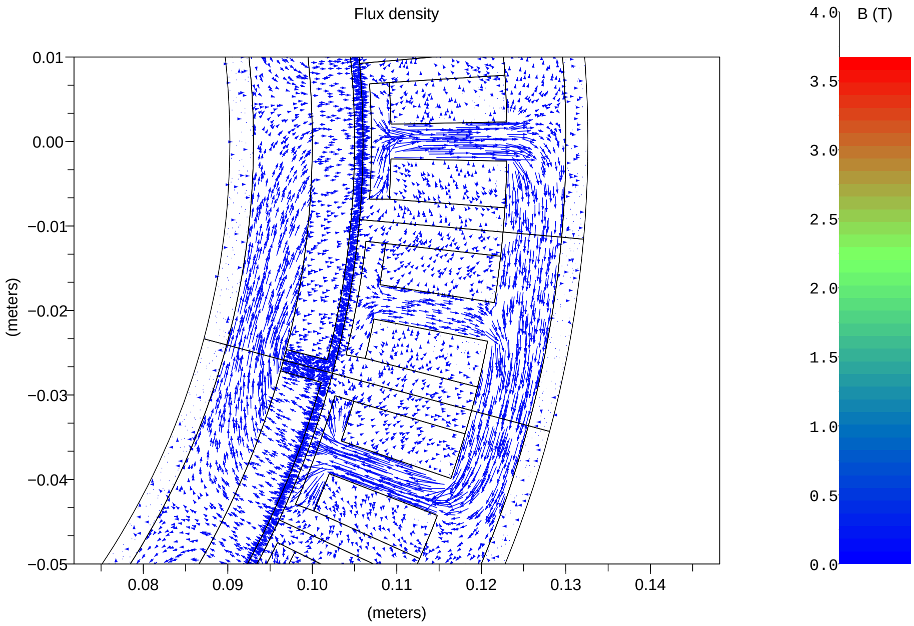

Figure 8.13: Flux directions from FEM calculations. The color scale is not valid.

Chapter 9

Results, conclusions

From the initial analysis of the load characteristics, anticipated speed of a Stortriss boat with this motor is about 5 knots. Different motor types were analyzed with the conclusion that a brushless permanent magnet motor was the best choice for this application.

This thesis has produced a complete set of parameters to build a brushless permanent motor for use for a propeller inside rotor boat motor. The dimension of the motor was first calculated analytically with help of Matlab. The resulting dimension of the motor was put into a finite element model to make sure the values from the analytical calculation were correct.

An important part of a permanent magnet motor is the position sensor. Different sensor types were considered and hall effect sensors were chosen.

In a trade off between iron core utilization and control complexity the optimal number of phases was considered to be three.

The number of magnets poles was chosen to 12 based on the desired speed of the motor in relation to an acceptable electrical commutation frequency.

For the selection of the number of slots two cases were evaluated, 27 slots and 36 slots design. It turned out that 36 slots gave a bit more torque and the difference in cogging torque was small. Therefore a 36 slots stator design was chosen.

For the combination of poles and slots that was chosen an algorithm was applied to obtain windings for the motor.

Test was made on ferrite magnet material that showed that it is possible to use ferrite magnets in salt sea water without coating. That is important since that reduce the water (air) gap.

The main physical characteristics of the designed motor can be seen in table 9.1. The main design constants can be studied in table 9.2.

Table 9.1: Important physical characteristics

P	T	U	I	δ	B_g
$[W]$	$[Nm]$	$[V]$	$[A]$	$[A/mm^2]$	$[T]$
2000	20	12	167	4.6	0.24

Table 9.2: Main design constants

Nr of poles	12
Nr of slots	36
Nr of turns	3
Rotor inner radius	9.3 cm
Rotor outer radius	10.5 cm
Stator inner radius	10.7 cm
Stator outer radius	13 cm
Axial length	6 cm

9.1 Future work

Next step is to build a prototype. With the prototype the design can be tested and to see if it behaves as the calculations predicts. An interesting parameter to study on the prototype is the power consumption during different situations, in terms of wind angle on the boat, waves and other factors that have an impact on the power consumption. From that test an estimation can be done on how far the boat can travel for a given battery capacity. It is also interesting to measure the efficiency of the motor itself. It might also be interesting to study the performance of the design as a generator when there is enough wind.

Besides the electrical characteristics, it is very interesting to study how this design works from a mechanical point of view. How hard it is to get the rotor-propeller part to move inside the stator.

Appendix A

Matlab code

Listing A.1: MATLAB design constants

```
%Torque
T = 20;

%Mecanical speed:
w_m = 1000 /60*2*pi;

% _Number of phases_
%Motors torque ripple is decreses as the number of phases increses. The materials in the
    motor, magnets, copper iron and so on can be more utilizated as the number of
    phases increses. The drawback with an increasing number of phases is that more
    transistors are needed to generate the current to the motor. In addition more wires
    are needed from the power electronic and around in the motor. The lowest number of
    phases, two needs only two power switches and only one hall switch. However the
    torque can be zero at some positions and special ?word? to get the motor rotating in
    disired direction. The two phase design is therefor only used in light-duty deisgn
    that have to be very cost effective, small fans for example. The most commonly done
    tradeoff and the one that is done in this design is to use three phese.
N_ph = 3;                  % Number of phases


%_Magnet poles_
%*The number of magnet poles should be inverely proportional to the maximum speed of
    rotation. *Every time the number of magnets poles is doubled the required thicknes
    of the rotor yoke or back-iron inside the magnets is reduced by one half. This is
    also the true for the stator yoke. The diameter decrese therefore then the number of
    magnet poles increases. I choose 100 Hz as the electrical frequency at nominal
    mecanical speed. From that, I can get an idea of how many magnets are needed. A test
    with 50 Hz gives substasional increase of the backirons. Higher frequencies gives
    higher core losses due to eddy currents.
f_e = 100;
w_e = 2*pi*f_e;
N_m_approx = w_e/w_m*2;  % = 12
N_m = 12;                  % Number of magnet poles

%disp('!!!!!!!!!!!! N_m')
%N_m = 6   % gives a valid winding acording to algorithm in Hendershot (but with long
    endturns)

%_Number of slots per pole per phase (N_spp)_
%The number of slots per pole per phase is choosen to 1.5. 1 may seems like a obvius
    choose. The advatage of 1.5 is to reduce the cogging torque. Cogging torque (
    reluctace torque) is an undisired torque that comes from the effect that the rotor
    magnets wants to align to the stator so they are close to as much stator iron as
    possible. A N_spp of 1.5 will cancel the cogging torque of each end of the magnets
    so the total cogging torque is close to zero. Another advantage of a N_spp of 1.5
    instead of 1 is that the back emf will be more sinusoial and smooth. Having a higher
    N_spp has the disadvantage that the utilazion of slot space usally is lower it
    there are many small slots comared to few bigger slots. In addition many slots
    consumes more work to wind. ????Check if 1.5 is the best value????

%N_spp = 1.5
disp('Test with N_spp = 1 !')
N_spp = 1
```

```
N_sp = N_spp * N_m;        % Number of slots per phase
N_s = N_sp * N_ph          % Number of slots
N_sm = N_spp * N_ph        % Number of slots per pole


% _Permanent Magnet properties_
        % see text for explination, ferrite, Y28

        % Recoil permeability of the material, meanvalue form manufacture:
        mu_R =  (1.05+1.3)/2

        % Magnet remanence, meanvalue form manufacture:
        B_r=(370e-3 + 400e-3)/2

%---------


g = 2E-3;                  % Air gap (m)


B_max = 1.5;               % Maximum steel flux density. 1.5 Hendershot, p. 3-24
                           % and indications form Sura cataloge
k_st = 0.8;                % Stacking factor: cross section occupied by
                           % magnetic material / total cross section, 0.5-0.95,
                           % test with 0.8
alpha_sd = 0.5             % Shoe depth fraction (d_1 + d_2)/w_tb

%Peak back emf:
E_max = 12;

%Conductor packing factor: Area occupied by conductor / total area. Typically k_cp is
     less than 50%. It can be know only through experience. My starting value is 0.4
k_cp = 0.4;

%Maximum allowable current density, J_max. This is limited only by how good the cooling
     of the motor is. For copper materials it's something between 4 and 10 MA/m^2. If J_c
      is higher than J_max the slot area must be increst. I set j_max to 10E6 A/m^2since
     the water should be able to cool the motor effecient.
J_max = 10E6;

%disp('J_max !!!!!!!!!!!!!!!!!!1 ')
%J_max = 10E7;
```

Listing A.2: MATLAB main code

```
clear global
clear all
clf

design_constants

R_so = 12E-2               % Outside stator radius

R_ro = 10.5E-2;            % Outside rotor radius (m)

l_m = 0.5E-2               % Magnet length in r direction


L = 2E-2                   % Axial length of motor


alpha_m = 0.95             % Magnet fraction: Magnet length
                           % / (Magnet length + space between magnets)

w_s = 5E-3;                % Slot opening


%---------


alpha_cp = floor(N_spp) / N_spp %Coil-pole fraction


theta_p = 2 * pi /N_m     %Angular pole pitch
theta_s = 2*pi / N_s      %Angluar slot pitch
theta_se = pi/N_sm        %Slot pitch, electrical radians

R_si = R_ro + g;          %Inside stator radius
tau_p = R_si * theta_p    %Pole pitch
tau_c = alpha_cp * tau_p  %Coil pitch
tau_s = R_si * theta_s    %Slot pitch ar air gap
```

```
w_t = tau_s - w_s         %Tooth width at air gap
k_d = sin(N_spp*theta_se/2) / ( N_spp * sin(theta_se/2)) %Distribution factor
k_p = alpha_cp            %Pitch factor
k_s = 1 - theta_se/(2*pi) %Skrew factor

C_phi = 2 * alpha_m / (1 + alpha_m) % Flux concentration factor
P_c = l_m / (g * C_phi) % Permeance coeffcient
if P_c < 4
        disp('-------␣Warning!␣P_c␣<␣4␣------------------')
end

k_ml = 1+4*l_m/(pi*mu_R*alpha_m*tau_p)*log(1+pi*g/((1-alpha_m)*tau_p)) %Magnet
                          % leakage factor
g_c = g + l_m/mu_R        % Effective air gap for Carter coefficient
k_c = (1-1/(tau_s/w_s*(5*g_c/w_s+1)))^-1 %Carter coefficient
A_g = tau_p*L*(1+alpha_m)/2 %Air gap area
B_g = C_phi/(1+mu_R*k_c*k_ml/P_c)*B_r %Air gap flux density
phi_g = B_g * A_g         %Air gap flux
w_bi = phi_g / (2*B_max * k_st * L) % Back iron width
w_tb = 2/N_sm*w_bi        % Tooth width
R_sb = R_so - w_bi        % Stator back iron radius
R_ri = R_ro - l_m - w_bi % Rotor inside radius
w_sb = R_sb*theta_s - w_tb % Slot botton width

w_si = (R_si+alpha_sd*w_tb)*theta_s - w_tb % Slot width inside shoes
alpha_s = w_si/(w_si+w_tb) % Slot fraction inside shoes
d_s = R_sb-R_ro-g         % Total slot depth
d_3 = d_s - alpha_sd*w_tb % Conductor slot depth
d_1_plus_d_2 = alpha_sd*w_tb %Shoe depth, split between d_1 and d_2




%Turns per slot:
n_s=floor(E_max / (N_m*k_d*k_p*k_s*B_g*L*R_ro*N_spp*w_m))

%Slot current:
I_s = T / (N_m*k_d*k_p*k_s*B_g*L*R_ro*N_spp)

%Phase currnet:
I_ph = I_s / (N_ph*n_s)


%linetype = 'r'
%clearplot
%plot(xcords(-w_t,w_t,R_si),ycords(0, 0,R_si),linetype)
%hold on
%plot(xcords(-w_t,-w_t,R_si),ycords(0, d_1_plus_d_2,R_si),linetype)
%plot(xcords( w_t, w_t,R_si),ycords(0, d_1_plus_d_2,R_si),linetype)
%plot(xcords(-w_t, -w_tb/2,R_si),ycords(d_1_plus_d_2, d_1_plus_d_2,R_si),linetype)
%plot(xcords( w_t,  w_tb/2,R_si),ycords(d_1_plus_d_2, d_1_plus_d_2,R_si),linetype)
%plot(xcords(-w_tb/2, -w_tb/2,R_si),ycords(d_1_plus_d_2, d_1_plus_d_2+d_3,R_si),linetype
    )
%plot(xcords( w_tb/2,  w_tb/2,R_si),ycords(d_1_plus_d_2, d_1_plus_d_2+d_3,R_si),linetype
    )
%plot(xcords( w_tb/2,  w_tb/2+w_sb,R_si),ycords(d_1_plus_d_2+d_3, d_1_plus_d_2+d_3,R_si)
     ,linetype)


disp(sprintf('To␣the␣.PROB␣file:'));
disp(sprintf('Devicelength␣␣␣␣␣%E',L));


figure(1)

nrofnodes = 0;
nroflines = 0;
nrofmaterialregions = 0;

global nodesfp;
global linesfp;
nodesfp = fopen('nodes.txt','w+');
linesfp = fopen('lines.txt','w+');
materialsfp = fopen('materials.txt','w+');

excitationfp = fopen('excitation','w+');

statormaterialcode = 10;
%lowerconductormaterialposcode = 11;
%upperconductormaterialposcode = 12;
```

```
%lowerconductormaterialnegcode = 13;
%upperconductormaterialnegcode = 14;

airmaterialcode = 100;
airgapmaterialcode = 101;

rotorbackironmaterialcode = 102;

magnetmaterialbasecode = 200;

slotmaterialbasecode = 300;

alt = 'g';

hold on
axis equal

save_node20 = NaN;
save_node19 = NaN;
save_node11 = NaN;
save_node12 = NaN;
first_node18 = NaN;
first_node1 = NaN;
first_node2 = NaN;
first_node21 = NaN;


%______________________Stator________


%20                  11----12        19
%                    16     22
%                    |      |
%                    |      |
%   7----8           15     |
%   |    |           |      |
%   |    9-----------10     |
%   |                       |
%   |                       |
%   |                       |
%   |    4-----------3      |
%   |    |           |      |
%   6----5           14     |
%                    |      |
%                    |      |
%                    13     17
%21                  2------1        18


%stator line type:
slt = 'r';

%arc(0, 2*pi, R_so, slt);     % only for plotting not for meshing!!!!!!!!!!!


% air line between stator and rotor:
%disp('skipping air line between stator and rotor:')
if 1
[air_nodes_1, nrofnodes, nroflines] = arc_core(0 ,pi, (R_ro+R_si)/2 , alt, NaN, NaN,
    nrofnodes, nroflines);
[air_nodes_2, nrofnodes, nroflines] = arc_core(pi ,2*pi, (R_ro+R_si)/2 , alt,
    air_nodes_1(end,1), air_nodes_1(1,1), nrofnodes, nroflines);


air_nodes = [air_nodes_1(1:end,:) ; air_nodes_2(2:end-1,:)];
current_air_node = air_nodes(1,1);
end

for slot = 0:    (N_s-1)

    %(All "lower" and "upper" terms refer to a slot placed at an angle ...
        %zero at the unit circle)

    %Angle base:
    ab = 2*pi / N_s * slot;

    %------------ lower part -------

    %floor of slot section - to simplify generation of infile to mesh generator

    [node1 , start_node, nrofnodes, nroflines] = radial_line(R_so, R_sb, ab+ -theta_s/2,
          slt, save_node12, save_node11, nrofnodes, nroflines);
    lower_floor_of_slot_section_end_node = start_node;
```

```
if isnan(first_node1)
    first_node1 = node1;
end

node2_x_cord = R_sb* cos(ab+ -theta_s/2);
node2_y_cord = R_sb* sin(ab+ -theta_s/2);



%inner stator part located below a line from origon through the ...
%end of shoe line. The inner stator line is divided in two parts ...
%to simplify the bulding of the conductor area.
[node2, node13, trash, start_node, nrofnodes, nroflines] = arc(ab+ -theta_s/2,ab+ -
    w_t/2 /R_si, R_sb, slt, start_node, NaN, nrofnodes, nroflines);
node14 = start_node;

if isnan(first_node2)
    first_node2 = node2;
end

%inner stator part located above a line from origon through the ...
%end of shoe line.
[lowerconductorstartnode,trash, trash, start_node, nrofnodes, nroflines] = ...
    arc(ab+ -w_t/2 /R_si,ab+ -w_tb/2 /R_sb, R_sb, slt, start_node, NaN, nrofnodes,
        nroflines);

%"leg":
%R_sb on the following line should have been R_si+d_1_plus_d_2. ...
%R_si+d_1_plus_d_2 is approximated with R_sb on every place there ...
%it is used to calculate the angle corsponding to w_tb/2. This is ...
%done to get the inner stator line fit exactly to the leg line and ...
%the line line to fit to the top of the shoe line. Otherwise ...
%this lines end and start points want be the same point. This can ...
%cause trubble during the mesh and fem process.
[trash, start_node, nrofnodes, nroflines] = radial_line(R_sb, R_si+d_1_plus_d_2,ab+
     -w_tb/2 /R_sb, slt, start_node, NaN, nrofnodes, nroflines);


%top of shoe:
%see note above regarding the use of R_sb on the following line
[trash,trash, trash, start_node, nrofnodes, nroflines] = arc(ab+ -w_tb/2 /R_sb,ab+
    -w_t/2 /(R_si+d_1_plus_d_2), (R_si+d_1_plus_d_2),slt, start_node, NaN,
    nrofnodes, nroflines);
lower_top_of_shoe_end_node = start_node;

%end of shoe:
[trash, start_node, nrofnodes, nroflines] = radial_line(R_si+d_1_plus_d_2, R_si, ab+
      -w_t/2 /R_si, slt, start_node, NaN, nrofnodes, nroflines);

%------------ both lower and upper part ------

%under shoe:
[trash,trash, trash, start_node, nrofnodes, nroflines] = arc(ab+ -w_t/2 /R_si,ab+
    w_t/2 /R_si, R_si, slt, start_node, NaN, nrofnodes, nroflines);

%------------ upper part ---------

%end of shoe:
[trash, start_node, nrofnodes, nroflines] = radial_line(R_si, R_si+d_1_plus_d_2,ab+
      w_t/2 /R_si, slt, start_node, NaN, nrofnodes, nroflines);

%top of shoe:
%see note above regarding the use of R_sb on the following line
[upper_top_of_shoe_start_node,trash, trash, start_node, nrofnodes, nroflines] = arc
    (ab+ w_t/2 /(R_si+d_1_plus_d_2),ab+  w_tb/2 /R_sb, (R_si+d_1_plus_d_2),slt,
    start_node, NaN, nrofnodes, nroflines);


%"leg"
%see note above regarding the use of R_sb on the following line
[trash, start_node, nrofnodes, nroflines] = radial_line(R_si+d_1_plus_d_2, R_sb,ab+
      w_tb/2  /R_sb, slt, start_node, NaN, nrofnodes, nroflines);


%inner stator part located below a line from origon through the ...
%end of shoe line. The inner stator line is divided in two parts ...
%to simplify the bulding of the conductor area.
[trash, trash, trash, start_node, nrofnodes, nroflines] = ...
    arc(ab+ +w_tb/2 /R_sb, ab+ +w_t/2 /R_si, R_sb, slt, start_node, NaN, nrofnodes
        , nroflines);
upperconductorendnode = start_node;
```

```
%inner stator part located above a line from origon through the ...
%end of shoe line.
if slot == N_s-1 %the last loop
    end_node = first_node2;
else
    end_node = NaN;
end

[node15, trash, node16, start_node, nrofnodes, nroflines] = arc(ab+ +w_t/2 /R_si,ab
     + +theta_s/2, R_sb, slt, start_node, end_node, nrofnodes, nroflines);
 node11 = start_node;

node11_x_cord = R_sb*cos(ab+ +theta_s/2);
node11_y_cord = R_sb*sin(ab+ +theta_s/2);


%top of a slot section - to simplify generation of infile to mesh generator
if slot == N_s-1 %the last loop
    end_node = first_node1;
else
    end_node = NaN;
end

[save_node11, start_node, nrofnodes, nroflines] =radial_line(R_sb, R_so, ab+ +
     theta_s/2, slt, start_node, end_node, nrofnodes, nroflines);
save_node12=start_node;

%------------ both lower and upper part ------

%Outer stator:
[node12, node22, trash, trash, nrofnodes, nroflines] = arc(ab+ +theta_s/2 , ab+ -
     theta_s/2, R_so, slt, start_node, node1, nrofnodes, nroflines);
node17 = nrofnodes;


%Lower conductor:
%fprintf(linesfp, '#Lower conductor\n');
for conductornode = lowerconductorstartnode+1 : lower_top_of_shoe_end_node
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, conductornode-1, conductornode);
end
nroflines = nroflines + 1;
fprintf(linesfp, '%i %i %i\n', nroflines, lower_top_of_shoe_end_node,
     lowerconductorstartnode);

%Upper conductor:
%fprintf(linesfp, '#Upper conductor\n');
for conductornode = upper_top_of_shoe_start_node+1 : upperconductorendnode
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, conductornode-1, conductornode);
end
nroflines = nroflines + 1;
fprintf(linesfp, '%i %i %i\n', nroflines, upperconductorendnode,
     upper_top_of_shoe_start_node);

%Air outside stator:
air_outside = 1.02;
%fprintf(linesfp, '#Air outside stator\n');

nroflines = nroflines + 1;
fprintf(linesfp, '%i %i %i\n', nroflines, node12, node22);

for node = node22+1 : node17
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, node-1, node);
end
nroflines = nroflines + 1;
fprintf(linesfp, '%i %i %i\n', nroflines, ...
    node17, node1);

if isnan(save_node19)
    nrofnodes = nrofnodes + 1;
    fprintf(nodesfp, '%i %E %E\n', nrofnodes, R_so*air_outside*cos(ab - ...
        theta_s/2), R_so*air_outside*sin(ab - theta_s/2) );
    node18 = nrofnodes;
else
    node18 = save_node19;
end

if isnan(first_node18)
    first_node18 = node18;
end
```

```
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, node1, node18);

    if slot == N_s-1 %the last loop
        end_node = first_node18;
    else
        end_node = NaN;
    end

    [trash, trash, trash, node19, nrofnodes, nroflines] = arc(ab+ ...
        -theta_s/2,ab+ +theta_s/2, R_so*air_outside, slt, node18, end_node, nrofnodes,
            nroflines);
    save_node19 = node19;

    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, node19, node12);


    %disp('skipping air inside stator')  ;
    make_air_inside_stator = 1;
    if make_air_inside_stator
%Air inside stator
    air_inside = 0.95;

    %fprintf(linesfp, '#Air inside stator\n');

    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', ...
        nroflines,node2 , node13);

    for node = node13 + 1 : node14
        nroflines = nroflines + 1;
        fprintf(linesfp, '%i %i %i\n', nroflines, node-1, node);
    end

    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', ...
        nroflines,node14 ,lower_top_of_shoe_end_node );

    for node = lower_top_of_shoe_end_node + 1 : upper_top_of_shoe_start_node
        nroflines = nroflines + 1;
        fprintf(linesfp, '%i %i %i\n', nroflines, node-1, node);
    end


    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', ...
        nroflines, upper_top_of_shoe_start_node, node15);

    for node = node15 + 1 : node16
        nroflines = nroflines + 1;
        fprintf(linesfp, '%i %i %i\n', nroflines, node-1, node);
    end

    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', ...
        nroflines, node16,node11 );

%      if slot == N_s-1 %the last loop
%     node20 = first_node21;
%      else
%     nrofnodes = nrofnodes + 1;
%     fprintf(nodesfp, '%i %E %E\n', nrofnodes, R_si*air_inside*cos(ab+ theta_s/2),
       R_si*air_inside*sin(ab+ theta_s/2) );
%     node20 = nrofnodes;
%      end

    % find node 20 located on the air between stator and rotor line:

    if (slot == N_s-1)
        node20 = first_node21;
    else
        old_distance = 10^10;
        distance = 10^10-1;

        while( distance < old_distance)
            old_distance = distance;
            distance = sqrt(( air_nodes(current_air_node,2) - node11_x_cord )^2 + ( ...;
                air_nodes(current_air_node,3) - node11_y_cord )^2);

            %plot([node11_x_cord  air_nodes(current_air_node,2)], [node11_y_cord
                 air_nodes(current_air_node,3)], 'k-')
```

```
            if current_air_node == air_nodes(end,1)
                current_air_node = air_nodes(1,1);
            else
                current_air_node = current_air_node + 1;
            end
        end

        if current_air_node-1 == air_nodes(1,1)
            node20 = air_nodes(end,1);
        else
            node20 = current_air_node -2;
        end

        %plot(air_nodes(node20,2), air_nodes(node20,3), 'ko')
    end

    %make line between node 11 and 20:
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines,node11 ,node20 );

    %if isnan(save_node20)
        %nrofnodes = nrofnodes + 1;
        %fprintf(nodesfp, '%i %E %E\n', nrofnodes, R_si*air_inside*cos(ab- theta_s/2),
            R_si*air_inside*sin(ab- theta_s/2) );
        %node21 = nrofnodes;
    %else
        %node21 = save_node20;
    %end

    %if isnan(first_node21)
%       first_node21 = node21;
%    end


    % find node 21 located on the air between stator and rotor line:

    if isnan(save_node20)
        old_distance = 10^10;
        distance = 10^10-1;

        while( distance < old_distance)
            old_distance = distance;
            distance = sqrt(( air_nodes(current_air_node,2) - node2_x_cord )^2 + ( ...
                air_nodes(current_air_node,3) - node2_y_cord )^2);

            if current_air_node == air_nodes(1,1)
                current_air_node = air_nodes(end,1);
            else
                current_air_node = current_air_node - 1;
            end
            %plot([node2_x_cord  air_nodes(current_air_node,2)], [node2_y_cord air_nodes
                (current_air_node,3)], 'k-')

        end

        if current_air_node+1 == air_nodes(end,1)
            node21 = air_nodes(1,1);
        else
            node21 = current_air_node + 2;
        end
    else
        node21 = save_node20;
    end

    if isnan(first_node21)
        first_node21 = node21;
    end

    % make line between node 20 and 21, (the nodes nummber in air_nodes
    % is decresing since we go clockwise)

    %plot(air_nodes(node21,2), air_nodes(node21,3), 'ro')

    node = node20-1 ;
    while (air_nodes(node,1) ~= node21)

        if node + 1 > air_nodes(end,1)
            start_node = air_nodes(1,1);
        else
            start_node = air_nodes(node + 1,1);
        end
        nroflines = nroflines + 1;
```

```
        fprintf(linesfp, '%i %i %i\n', nroflines, start_node, air_nodes(node,1));
        %disp( sprintf( '%i %i %i', nroflines, start_node, ...
            %air_nodes(node,1)) );

        if node == air_nodes(1,1)
            node = air_nodes(end,1);
        else
            node = node -1;
        end
    end
    if node + 1 > air_nodes(end,1)
        start_node = air_nodes(1,1);
    else
        start_node = air_nodes(node + 1,1);
    end
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, start_node, air_nodes(node,1));


    %line between node 21 and 2:

    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines,node21 , node2);

    save_node20 = node20;
    end %skip air inside stator

    %------------- Material assignment points ----------

    %Stator:
    nrofmaterialregions = nrofmaterialregions+1;
    fprintf(materialsfp, '%i %E %E %i\n', nrofmaterialregions, ...
        (R_so+R_si)/2 * cos(ab),(R_so+R_si)/2 * sin(ab) ...
        ,statormaterialcode); %the %E and %E gives x and y cordinats ...
        %for a point in the middle of the stator segment
    %for human verifying:
    plot((R_so+R_si)/2 * cos(ab),(R_so+R_si)/2 * sin(ab),'x')


    % Conductors

%     if mod(slot,2) == 0
%         lowerconductormaterialcode = lowerconductormaterialposcode;
%         upperconductormaterialcode = upperconductormaterialposcode;
%     else
%         lowerconductormaterialcode = lowerconductormaterialnegcode;
%         upperconductormaterialcode = upperconductormaterialnegcode;
%     end

if slot == 0
        lowerconductormaterialcode = slotmaterialbasecode + (N_s * 2) ...
            - 1;
        upperconductormaterialcode = slotmaterialbasecode;
else
        lowerconductormaterialcode = slotmaterialbasecode + (slot * 2) - 1;
        upperconductormaterialcode = slotmaterialbasecode + (slot * 2);
end
    %Lower conductor:

    distancefromorigon = ((R_si+d_1_plus_d_2) + R_sb)/2;

    nrofmaterialregions = nrofmaterialregions+1;
    fprintf(materialsfp, '%i %E %E %i\n', nrofmaterialregions, ...
        distancefromorigon * cos(ab -(w_tb/2+w_t/2)/2/distancefromorigon),...
        distancefromorigon * sin(ab -(w_tb/2+w_t/2)/2/distancefromorigon) ...
        ,lowerconductormaterialcode);
    %for human verifying:
    plot(distancefromorigon * cos(ab -(w_tb/2+w_t/2)/2/distancefromorigon),...
        distancefromorigon * sin(ab -(w_tb/2+w_t/2)/2/distancefromorigon),'o');

    %Upper conductor:
    nrofmaterialregions = nrofmaterialregions+1;
    fprintf(materialsfp, '%i %E %E %i\n', nrofmaterialregions, ...
        distancefromorigon * cos(ab +(w_tb/2+w_t/2)/2/distancefromorigon),...
        distancefromorigon * sin(ab +(w_tb/2+w_t/2)/2/distancefromorigon) ...
        ,upperconductormaterialcode);
    %for human verifying:
    plot(distancefromorigon * cos(ab +(w_tb/2+w_t/2)/2/distancefromorigon),...
         distancefromorigon * sin(ab +(w_tb/2+w_t/2)/2/distancefromorigon),'o');


    %Air outside stator:
```

```
    nrofmaterialregions = nrofmaterialregions+1;
    fprintf(materialsfp, '%i %E %E %i\n', nrofmaterialregions, ...
        (R_so + R_so*air_outside)/2 *cos(ab), (R_so + R_so*air_outside)/2*sin(ab),
            airmaterialcode);
    plot((R_so + R_so*air_outside)/2*cos(ab), (R_so + R_so*air_outside)/2*sin(ab),'o');

    %Air inside stator, i.e. air in the airgap on the stator side of
    %the airgap airline.:
    if make_air_inside_stator

    nrofmaterialregions = nrofmaterialregions+1;
    fprintf(materialsfp, '%i %E %E %i\n', nrofmaterialregions, ...
          ((R_ro+R_si)/2 + R_si)/2*cos(ab), ((R_ro+R_si)/2 + R_si)/2*sin(ab),
              airmaterialcode);
    %for human verifying:
    %plot( ((R_ro+R_si)/2 + R_si)/2*cos(ab), ((R_ro+R_si)/2 + R_si)/2*sin(ab),'ok');
    end %air inside stator

    %disp('pause')
    %pause

 end


%___________________ Rotor __________________

if 0
disp('------------------ Skipping rotor -------------------')
else

current_air_node = air_nodes(1,1);


%rotor line type:
rlt = 'b';
%rotor line type air
rlt_air = 'y';

%Outer line of rotor
%arc(0, 2*pi, R_ro,rlt);


%Inner line of magnets
%arc(0, 2*pi, R_ro-l_m,rlt);


%Inner line of rotor
%arc(0, 2*pi, R_ri,rlt);


%for magnet_pole = 0:(N_m-1)
         %Angle base:
%        ab = theta_p * magnet_pole;

         %magnet
%        radial_line(R_ro-l_m, R_ro, ab+ -theta_p/2*alpha_m    , rlt)
%        radial_line(R_ro-l_m, R_ro, ab+ +theta_p/2*alpha_m    , rlt)

%end

%    r22---r12-----r11--------r23
%    r21    r13      r10           |
%    |      |        |             |
%    |      |        r9            |
%    |      |        r5-----r6     |
%    |      |        r4     r7     |
%    |      |        |      |      |
%    |  A   |   B    |  M   |  A   |
%    |  i   |   a    |  a   |  i   |
%    |  r   |   c    |  g   |  r   |
%    |      |   k    |  n   |  g   |
%    |      |        |  e   |  a   |
%    |      |   i    |  t   |  p   |
%    |      |   r    |      |      |
%    |      |   o    |      |      |
%    |      |   n    |      |      |
%    |      |        |      |      |
%    |      |        r3     r8     |
%    |      |        r2-----r1     |
%    |      |        r18           |
%    |      |        |             |
%    r20    r14      r17           |
%    r19-----r15-----r16-------r24
```

```
first_noder16 = NaN;
first_noder15 = NaN;
first_noder22 = NaN;
first_noder24 = NaN;
save_noder11 = NaN;
save_noder12 = NaN;
save_noder19 = NaN;
save_noder23 = NaN;

for magnet_pole = 0 :    (N_m-1)
    %Angle base:
    ab = theta_p * magnet_pole;

    %magnet:
    [noder1,noder2,nrofnodes,nroflines]=radial_line(R_ro, R_ro-l_m,   ab+  -theta_p/2*
        alpha_m, rlt, NaN, NaN ,nrofnodes,nroflines);

    [trash, noder3, noder4, noder5, nrofnodes, nroflines] = arc(ab+ ...
        -theta_p/2*alpha_m ,ab+ +theta_p/2*alpha_m, R_ro-l_m, rlt, noder2, NaN,
            nrofnodes, nroflines);

    %%%%%%%%%%%%%[trash,noder6]=radial_line(R_ro-l_m, R_ro, ab+ ...
    %%%%%%%%%%%%          +theta_p/2*alpha_m , rlt, noder5, NaN,nrofnodes, nroflines);

    [trash,noder6,nrofnodes,nroflines]=radial_line(R_ro-l_m, R_ro, ab+ ...
           +theta_p/2*alpha_m , rlt, noder5, NaN,nrofnodes, nroflines);

        [trash, noder7, noder8, trash, nrofnodes, nroflines] = arc(ab+ ...
        +theta_p/2*alpha_m ,ab+ -theta_p/2*alpha_m      ...
        , R_ro, rlt, noder6, noder1, nrofnodes, nroflines);

    %fprintf(linesfp,'--- 1 ---\n');

    %rotor back iron:

    [noder16, noder17, noder18, trash, nrofnodes, nroflines] = arc(ab+ ...
        -theta_p/2, ab+  -theta_p/2*alpha_m , R_ro-l_m, rlt, save_noder11, noder2,
            nrofnodes, nroflines);


     noder16_x_cord = (R_ro-l_m) * cos(ab+ -theta_p/2);
     noder16_y_cord = (R_ro-l_m) * sin(ab+ -theta_p/2);

    %fprintf(linesfp,'--- 2 ---\n');


    for node = noder2 + 1 : noder5
        nroflines = nroflines + 1;
        fprintf(linesfp, '%i %i %i\n', nroflines, node-1, node);
    end

    if magnet_pole == N_m-1
        end_node = first_noder16;
    else
        end_node = NaN;
    end

    [trash, noder9, noder10, noder11, nrofnodes, nroflines] = arc(ab+ ...
        +theta_p/2*alpha_m ,ab+ +theta_p/2, R_ro-l_m, rlt, noder5, end_node, nrofnodes,
             nroflines);

    noder11_x_cord = (R_ro-l_m) * cos(ab+  +theta_p/2);
    noder11_y_cord = (R_ro-l_m) * sin(ab+  +theta_p/2);


    if magnet_pole == N_m-1
        end_node = first_noder15;
    else
        end_node = NaN;
    end

    [trash,noder12,nrofnodes,nroflines]=radial_line(R_ro-l_m, R_ri, ab+ ...
           +theta_p/2 , rlt, noder11, end_node, nrofnodes, nroflines);

    [trash, noder13, noder14, noder15, nrofnodes, nroflines] = arc(ab+ ...
        +theta_p/2 ,ab+ -theta_p/2, R_ri, rlt, noder12, save_noder12, nrofnodes,
            nroflines);

    [trash,trash,nrofnodes,nroflines]=radial_line(R_ri, R_ro-l_m, ab+ ...
        -theta_p/2 , rlt, noder15, noder16, nrofnodes, nroflines);
```

```
% --- Air inside rotor ---

air_inside_rotor = 0.97;
[trash,noder22,nrofnodes,nroflines]=radial_line(R_ri, R_ri*air_inside_rotor, ab+ ...
    -theta_p/2 , rlt_air, noder15, save_noder19, nrofnodes, nroflines);

if magnet_pole == N_m-1
    end_node = first_noder22;
else
    end_node = NaN;
end

[trash, noder21, noder20, noder19, nrofnodes, nroflines] = arc(ab+ ...
    -theta_p/2 ,ab+ +theta_p/2, R_ri*air_inside_rotor, rlt_air, ...
    noder22, end_node, nrofnodes, nroflines);

[trash,noder12,nrofnodes,nroflines]=radial_line(R_ri*air_inside_rotor, R_ri, ab+ ...
    +theta_p/2 , rlt_air, noder19, noder12, nrofnodes, nroflines);

nroflines = nroflines + 1;
fprintf(linesfp, '%i %i %i\n', nroflines, noder12, noder13);

for node = noder13 + 1 : noder14
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, node-1, node);
end

nroflines = nroflines + 1;
fprintf(linesfp, '%i %i %i\n', nroflines, noder14, noder15);

% ----- Air outside rotor -----

nroflines = nroflines + 1;
fprintf(linesfp, '%i %i %i\n', nroflines, noder16, noder17);

for node = noder17 + 1 : noder2
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, node-1, node);
end

nroflines = nroflines + 1;
fprintf(linesfp, '%i %i %i\n', nroflines, noder2, noder1);

nroflines = nroflines + 1;
fprintf(linesfp, '%i %i %i\n', nroflines, noder1, noder8);

for node = noder8 - 1 : -1 : noder6
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, node+1, node);
end

nroflines = nroflines + 1;
fprintf(linesfp, '%i %i %i\n', nroflines, noder6, noder5);

for node = noder5 + 1 : noder10
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, node-1, node);
end

nroflines = nroflines + 1;
fprintf(linesfp, '%i %i %i\n', nroflines, noder10, noder11);

%---


% find node 23 located on the air between stator and rotor line:

if (magnet_pole == N_m-1)
    noder23 = first_noder24;
else
    old_distance = 10^10;
    distance = 10^10-1;

    while( distance < old_distance)
        old_distance = distance;
        distance = sqrt(( air_nodes(current_air_node,2) - noder11_x_cord )^2 + (
            ...;
            air_nodes(current_air_node,3) - noder11_y_cord )^2);

        %plot([noder11_x_cord air_nodes(current_air_node,2)], [noder11_y_cord
            air_nodes(current_air_node,3)], 'k-')

        if current_air_node == air_nodes(end,1)
```

```
                current_air_node = air_nodes(1,1);
            else
                current_air_node = current_air_node + 1;
            end
        end

        if current_air_node-1 == air_nodes(1,1)
            noder23 = air_nodes(end,1);
        else
            noder23 = current_air_node -2;
        end

        %plot(air_nodes(noder23,2), air_nodes(noder23,3), 'ko')
    end

    %make line between node r11 and r23:
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, noder11 ,noder23 );

    %if isnan(save_node20)
        %nrofnodes = nrofnodes + 1;
        %fprintf(nodesfp, '%i %E %E\n', nrofnodes, R_si*air_inside*cos(ab- theta_s/2),
             R_si*air_inside*sin(ab- theta_s/2) );
        %node21 = nrofnodes;
    %else
        %node21 = save_node20;
    %end

    %if isnan(first_node21)
%        first_node21 = node21;
%    end


   % find node 24 located on the air between stator and rotor line:

    if isnan(save_noder23)
        old_distance = 10^10;
        distance = 10^10-1;

        while( distance < old_distance)
            old_distance = distance;
            distance = sqrt(( air_nodes(current_air_node,2) - noder16_x_cord )^2 + ( ...
                air_nodes(current_air_node,3) - noder16_y_cord )^2);

            if current_air_node == air_nodes(1,1)
                current_air_node = air_nodes(end,1);
            else
                current_air_node = current_air_node - 1;
            end
            %plot([noder16_x_cord  air_nodes(current_air_node,2)], [noder16_y_cord
                 air_nodes(current_air_node,3)], 'k-')

        end

        if current_air_node+1 == air_nodes(end,1)
            noder24 = air_nodes(1,1);
        else
            noder24 = current_air_node + 2;
        end
    else
        noder24 = save_noder23;
    end

    if isnan(first_noder24)
        first_noder24 = noder24;
    end

    % make line between node r23 and r24, (the nodes nummber in air_nodes
    % is decresing since we go clockwise)

    %plot(air_nodes(noder24,2), air_nodes(noder24,3), 'ro')

    node = noder23-1 ;
    while (air_nodes(node,1) ~= noder24)

        if node + 1 > air_nodes(end,1)
            start_node = air_nodes(1,1);
        else
            start_node = air_nodes(node + 1,1);
        end
        nroflines = nroflines + 1;

        fprintf(linesfp, '%i %i %i\n', nroflines, start_node, air_nodes(node,1));
```

```
        %disp( sprintf( '%i %i %i', nroflines, start_node, ...
            %air_nodes(node,1)) );

        if node == air_nodes(1,1)
            node = air_nodes(end,1);
        else
            node = node -1;
        end
    end
    if node + 1 > air_nodes(end,1)
        start_node = air_nodes(1,1);
    else
        start_node = air_nodes(node + 1,1);
    end
    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines, start_node, air_nodes(node,1));


    %line between node r24 and r16:

    nroflines = nroflines + 1;
    fprintf(linesfp, '%i %i %i\n', nroflines,noder24 , noder16);

    save_noder23 = noder23;
%-------------------

    %------------ Material assignment points ----------

    %Rotor back iron:

    nrofmaterialregions = nrofmaterialregions+1;
    fprintf(materialsfp, '%i %E %E %i\n', nrofmaterialregions, ...
        (R_ri + R_ro-l_m)/2*cos(ab), (R_ri + R_ro-l_m)/2*sin(ab),
            rotorbackironmaterialcode);
    %for human verifying:
    %plot( (R_ri + R_ro-l_m)/2*cos(ab), (R_ri + R_ro-l_m)/2*sin(ab),'ok');

    %Magnet:

    nrofmaterialregions = nrofmaterialregions+1;
    fprintf(materialsfp, '%i %E %E %i\n', nrofmaterialregions, ...
        (R_ro + R_ro-l_m)/2*cos(ab), (R_ro + R_ro-l_m)/2*sin(ab), magnetmaterialbasecode
            +magnet_pole);
    %for human verifying:
    %plot( (R_ro + R_ro-l_m)/2*cos(ab), (R_ro + R_ro-l_m)/2*sin(ab),'ok');

    % To the .PROB file:
    magnet_magnitude = (-1)^magnet_pole * B_r;
%     disp(sprintf('%i      B       [%E,    %E   0, 0]', ...
    fprintf(excitationfp, '%i     B       [%E,   %E,   0]\n', ...
        magnetmaterialbasecode+magnet_pole , magnet_magnitude * cos(ab),
            magnet_magnitude * sin(ab));

    % Air inside rotor:

    nrofmaterialregions = nrofmaterialregions+1;
    fprintf(materialsfp, '%i %E %E %i\n', nrofmaterialregions, ...
        (R_ri*air_inside_rotor + R_ri)/2*cos(ab), (R_ri*air_inside_rotor + R_ri)/2*sin(
            ab), airmaterialcode);
    %for human verifying:
    %plot((R_ri*air_inside_rotor + R_ri)/2*cos(ab), (R_ri*air_inside_rotor + R_ri)/2*sin
        (ab),'ok');


    %Air outside rotor, i.e. air in the airgap on the rotor side of
    %the airgap airline.:
    nrofmaterialregions = nrofmaterialregions+1;
    fprintf(materialsfp, '%i %E %E %i\n', nrofmaterialregions, ...
        ((R_ro+R_si)/2 + R_ro)/2*cos(ab), ((R_ro+R_si)/2 + R_ro)/2*sin(ab),
            airgapmaterialcode);
    %for human verifying:
    %plot( ((R_ro+R_si)/2 + R_ro)/2*cos(ab), ((R_ro+R_si)/2 + R_ro)/2*sin(ab),'ok');


   disp(sprintf('To the .PROB file:'));

    %disp(sprintf('%i [0., 0., 0.] Cylinder [0., 0., 1.] %E [0., 0., 0.] [0., 0., 0.] ',
        ...
        %nrofmaterialregions,...
        %((R_ro+R_si)/2 + R_ro)/2));
    disp(sprintf('%i [0., 0., 0.] Cylinder [0., 0., 1.] %E [0., 0., 0.] [0., 0., 0.] ',
        ...
        airgapmaterialcode,...
```

```
        ((R_ro+R_si)/2 + R_ro)/2));
    disp(sprintf('There should only be one torque line in the .PROB-file , even if it is 
         repeted above.'));


    save_noder11 = noder11;
    save_noder12 = noder12;
    save_noder19 = noder19;

    if isnan(first_noder16)
        first_noder16 = noder16;
    end
    if isnan(first_noder15)
        first_noder15 = noder15;
    end
    if isnan(first_noder22)
        first_noder22 = noder22;
    end
end
end %skipping rotor


if 0
    disp('Skipping---------------------------------------------------')
else



% ---------------------- Plot of Windings ----------------------

%       1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
    28 29 30 31 32 33 34 35 36
p1 = [+1 +1 +1 -1 -1 -1 +1 +1 +1 -1 -1 -1 +1 +1 +1 -1 -1 -1 +1 +1 +1 -1 -1 -1 +1 +1 +1
    -1 -1 -1 +1 +1 +1 -1 -1 -1];
p2 = [-1 -1 +1 +1 +1 -1 -1 -1 +1 +1 +1 -1 -1 -1 +1 +1 +1 -1 -1 -1 +1 +1 +1 -1 -1 -1 +1
    +1 +1 -1 -1 -1 +1 +1 +1 -1];
p3 = [+1 -1 -1 -1 +1 +1 +1 -1 -1 -1 +1 +1 +1 -1 -1 -1 +1 +1 +1 -1 -1 -1 +1 +1 +1 -1 -1
    -1 +1 +1 +1 -1 -1 -1 +1 +1];

%Winding start radius:
wsr = (R_si+R_sb)/2;

%Winding start angle
wsa =  (w_tb+w_sb)/2 /R_sb;

%Winding ending angle:
wea = wsa + (w_sb + w_tb) / R_sb;

for phase = 1:3

        switch phase
        case 1
                p = p1;
                plt = 'm';
                %Winding radius:
                wr = R_so * 1.10; % times 1.1 so the winding will be ploted 10% from the
                        stator
        case 2
                p = p2;
                plt = 'g';
                %Winding radius:
                wr = R_so * 1.15;
        case 3
                p = p3;
                plt = 'c';
                %Winding radius:
                wr = R_so * 1.20;
        end

for slot = 1:N_s

        %previous slot:
        ps = slot - 1;
        if (ps < 1)
                ps = N_s;
        end

        %current slot:
        cs = slot;

        %next slot:
        ns = slot + 1;
        if (ns > N_s)
```

```
                ns = 1;
        end


        %Angle base:
        ab = 2*pi / N_s * (slot-1);


        % ------ Positive coil: ----------
        wsa_temp = wsa;
        wea_temp = wea;

        if (p(cs) > 0)  & (p(ps) < 1) %start of coil
                wsa_temp = wsa_temp*1.2;
                radial_line_plot_only(wsr, wr, ab+wsa_temp, plt)
        end

        if (p(cs) > 0)  & (p(ns) < 1) %end of coil
                wea_temp = wea_temp*0.8;
                radial_line_plot_only(wsr, wr, ab+wsa_temp, plt)
        end

        if (p(cs) > 0)  & (p(ns) > 0) %continue coil
                arc_plot_only(ab+wsa_temp, ab+wea_temp , wr, plt)
                radial_line_plot_only(wr*0.98, wr*1.02, ab+(wsa+wea)/2, plt)
        end

        % ------ Negative coil: ----------
        wsa_temp = wsa;
        wea_temp = wea;

        if (p(cs) < 0)  & (p(ps) > -1) %start of coil
                wsa_temp = wsa_temp*1.2;
                radial_line_plot_only(wsr, wr, ab+wsa_temp, plt)
        end

        if (p(cs) < 0)  & (p(ns) > -1) %end of coil
                wea_temp = wea_temp*0.8;
                radial_line_plot_only(wsr, wr, ab+wsa_temp, plt)
        end

        if (p(cs) < 0)  & (p(ns) < 0) %continue coil
                arc_plot_only(ab+wsa_temp, ab+wea_temp , wr, plt)
                arc_plot_only(ab+(wsa+wea)/2 - (wsa+wea)/2 * 0.1, ab+(wsa+wea)/2 + (wsa+
                    wea)/2 * 0.1, wr * 1.02,plt)
        end

end % slot
end % phase

if 0

%---------------------------------------- Start of test
    ----------------------------------------------------

p1 = p2; %----------------------------

plt = 'b'; %----------------------------

%Winding start radius:
wsr = (R_si+R_sb)/2;

%Winding radius:
wr = R_so * 1.15; % times 1.1 so the winding will be ploted 10% from the stator
    %-------------------------

%Winding start angle
wsa =  (w_tb+w_sb)/2 /R_sb;

%Winding ending angle:
wea = wsa + (w_sb + w_tb) / R_sb;

for slot = 1:N_s

        %previous slot:
        ps = slot - 1;
        if (ps < 1)
                ps = N_s;
        end

        %current slot:
        cs = slot;
```

```
        %next slot:
        ns = slot + 1;
        if (ns > N_s)
                ns = 1;
        end


        %Angle base:
        ab = 2*pi / N_s * (slot-1);


        % ------ Positive coil: ----------

        if (p1(cs) > 0)  & (p1(ps) < 1) %start of coil
                radial_line_plot_only(wsr, wr, ab+wsa*1.2, plt)
        end

        if (p1(cs) > 0)  & (p1(ns) < 1) %end of coil
                radial_line_plot_only(wsr, wr, ab+wsa*0.8, plt)
        end

        if (p1(cs) > 0)  & (p1(ns) > 0) %continue coil
                arc_plot_only(ab+wsa, ab+wea , wr, plt)
                radial_line_plot_only(wr*0.98, wr*1.02, ab+(wsa+wea)/2, plt)
        end

        % ------ Negative coil: ----------

        if (p1(cs) < 0)  & (p1(ps) > -1) %start of coil
                radial_line_plot_only(wsr, wr, ab+wsa*1.2, plt)
        end

        if (p1(cs) < 0)  & (p1(ns) > -1) %end of coil
                radial_line_plot_only(wsr, wr, ab+wsa*0.8, plt)
        end

        if (p1(cs) < 0)  & (p1(ns) < 0) %continue coil
                arc_plot_only(ab+wsa, ab+wea , wr, plt)
                arc_plot_only(ab+(wsa+wea)/2 - (wsa+wea)/2 * 0.1, ab+(wsa+wea)/2 + (wsa+
                    wea)/2 * 0.1, wr * 1.02,plt)
        end
end

%------------------------------------ p 3 ----

p1 = p3; %----------------------------

plt = 'r'; %----------------------------

%Winding start radius:
wsr = (R_si+R_sb)/2;

%Winding radius:
wr = R_so * 1.2; % times 1.1 so the winding will be ploted 10% from the stator
    %---------------------------

%Winding start angle
wsa =  (w_tb+w_sb)/2 /R_sb;

%Winding ending angle:
wea = wsa + (w_sb + w_tb) / R_sb;

for slot = 1:N_s

        %previous slot:
        ps = slot - 1;
        if (ps < 1)
                ps = N_s;
        end

        %current slot:
        cs = slot;

        %next slot:
        ns = slot + 1;
        if (ns > N_s)
                ns = 1;
        end


        %Angle base:
        ab = 2*pi / N_s * (slot-1);
```

```
        % ------ Positive coil: ----------

        if (p1(cs) > 0) & (p1(ps) < 1) %start of coil
                radial_line_plot_only(wsr, wr, ab+wsa*1.2, plt)
        end

        if (p1(cs) > 0) & (p1(ns) < 1) %end of coil
                radial_line_plot_only(wsr, wr, ab+wsa*0.8, plt)
        end

        if (p1(cs) > 0) & (p1(ns) > 0) %continue coil
                arc_plot_only(ab+wsa, ab+wea , wr, plt)
                radial_line_plot_only(wr*0.98, wr*1.02, ab+(wsa+wea)/2, plt)
        end

        % ------ Negative coil: ----------

        if (p1(cs) < 0) & (p1(ps) > -1) %start of coil
                radial_line_plot_only(wsr, wr, ab+wsa*1.2, plt)
        end

        if (p1(cs) < 0) & (p1(ns) > -1) %end of coil
                radial_line_plot_only(wsr, wr, ab+wsa*0.8, plt)
        end

        if (p1(cs) < 0) & (p1(ns) < 0) %continue coil
                arc_plot_only(ab+wsa, ab+wea , wr, plt)
                arc_plot_only(ab+(wsa+wea)/2 - (wsa+wea)/2 * 0.1, ab+(wsa+wea)/2 + (wsa+
                     wea)/2 * 0.1, wr * 1.02,plt)
        end
end


%
    ------------------------------------------------------------------------------------------
end

end %skipping


% Current exitations to the .PROB file:


%       1      2      3      4      5      6      7      8      9      10     11     12     13     14
         15     16     17     18     19     20     21     22     23     24     25     26     27     28
         29     30     31     32     33     34     35     36
%       f   s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f
     s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f
     s   f   s   f   s   f   s   f   s   f   s   f   s   f   s   f   s
p1 = [ 0 -1  0  0 +1  0  0 +1  0  0 -1  0  0 -1  0  0 +1  0  0 +1  0  0 -1  0  0 -1  0
     0 +1  0  0 +1  0  0 -1  0  0 -1  0  0 +1  0  0 +1  0  0 -1  0  0 -1  0  0 +1  0  0
     +1  0  0 -1  0  0 -1  0  0 +1  0  0 +1  0  0 -1  0 ];
p2 = [ 0  0 -1  0  0 -1  0  0 +1  0  0 +1  0  0 -1  0  0 -1  0  0 +1  0  0 +1  0  0 -1
     0  0 -1  0  0 +1  0  0 +1  0  0 -1  0  0 -1  0  0 +1  0  0 +1  0  0 -1  0  0 -1  0
     0 +1  0  0 +1  0  0 -1  0  0 -1  0  0 +1  0  0 +1 ];
p3 = [+1  0  0 +1  0  0 -1  0  0 -1  0  0 +1  0  0 +1  0  0 -1  0  0 -1  0  0 +1  0  0
     +1  0  0 -1  0  0 -1  0  0 +1  0  0 +1  0  0 -1  0  0 -1  0  0 +1  0  0 +1  0  0 -1
      0  0 -1  0  0 +1  0  0 +1  0  0 -1  0  0 -1  0  0 ];

% state:     1  2  3  4  5  6
states = [ +1 +1  0 -1 -1  0 ; ...    % phase 1
           -1  0 +1 +1  0 -1 ; ...    % phase 2
            0 -1 -1  0 +1 +1];        % phase 3
state = 1;
current = p1 * states(1, state) + p2 * states(2, state) + p3 * states(3, state);

for half_slot = 0 : N_s *2 - 1

        fprintf(excitationfp, '%i     J       [0,   0,   %E]\n', ...
        slotmaterialbasecode+half_slot , current(half_slot+1)*1e6   );
        %                                                   +1 because matlab index of
              vectors starts at 1
end


fclose('all');

%----- make .poly file from nodes, lines and materials files ---------
polyfp = fopen('radial.poly','w+');
```

```
% nrofnodes nodes, 2 dimensions, no attributes, no boundary marker:
fprintf(polyfp,'%i 2 0 0\n', nrofnodes);

nodesfp = fopen('nodes.txt','r');
nodes  = fscanf(nodesfp,'%i %E %E',[3 inf]);
fprintf(polyfp,'%i %E %E\n',nodes);

fprintf(polyfp,'\n');

%nfoflines lines, no boundary marker:
fprintf(polyfp,'%i 0\n', nroflines);

linesfp = fopen('lines.txt','r');
lines  = fscanf(linesfp,'%i %i %i',[3 inf]);
fprintf(polyfp,'%i %i %i\n',lines);

fprintf(polyfp,'\n');

% hole in the middle:
fprintf(polyfp,'1\n');  % <---- one hole
fprintf(polyfp,'1  0  0\n');
%disp('Hole in air between rotor and stator')
%fprintf(polyfp,'2  %E  %E\n', ((R_ro+R_si)/2 + R_ro)/2, 0);
%plot( ((R_ro+R_si)/2 + R_ro)/2, 0,'or');

fprintf(polyfp,'\n');

%nrofmaterialregions Regional attributes (materials):
fprintf(polyfp,'%i\n',nrofmaterialregions);

materialsfp = fopen('materials.txt','r');
materials  = fscanf(materialsfp,'%i %E %E %i',[4 inf]);

maximum_area = ones(1, length(materials)) *   0.000002;

fprintf(polyfp,'%i %E %E %i %E\n',[materials' maximum_area']');

fclose('all');
hold off
```

Bibliography

[1] Morton Ray *Performance Electric Boats: Electric Boats for General Waters* `<http://www.rayeo.com/art_3.htm>` (2005-07-12)

[2] Peter Rye *Of Boats and Waves* <http://members.iinet.net.au/~bluep/wavedrag.html> (2005-07-12)

[3] T.J.E. Miller (2004) "Switched-reluctance machines", in *Handbook of electric motors*, 2:ed, edited Hamid A. Toliyat, Gerald B. Kliman, Marcel Dekker, Inc, New York

[4] Prakashraj Kasinathan (2003) *Integrated In-Wheel Motors for Low Power Traction Application*,Chalmers University of Technology, p 27

[5] Irving Gottlieb (1997) *Practical Electric Motor Handbook*, Newnes, Oxford, p. 63

[6] Essam S. Hamdi (1994) *Design of small electric machines*, John Wiley & Sons Ltd., West Sussex, England, p. 167

[7] Essam Hamdi (2003) *Permanent magnet and variable-reluctance drive systems*, Chalmers Reproservice, Göteborg, Sweden, p. 64

[8] J.R. Hendershot Jr., TJE Miller (1994) *Design of Brushless Permanent-Magnet Motors*, Magna Physics, Hillsboro, USA& Oxford University, New York, USA

[9] Duane C. Hanselman (1994) *Brushless permanent-magnet motors design*, McGraw-Hill, Inc., United States of America

[10] G. Qishan, G. Hongzhan (1985) *Effect of Slottning in PM Electric Machines*, Electric Machines and Power Systems, vol 10, pp 273-284