Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

Øyvind Sommer Klyve

Magnetic Modelling of Saturated IPMSMs, for Improved Torque Estimation and Accurate MTPA Control

Master's thesis in Energy and Environmental Engineering
Supervisor: Robert Nilssen
Co-supervisor: Roy Nilsen, Aravinda Perera
June 2021

NTNU
Norwegian University of Science and Technology

Øyvind Sommer Klyve

Magnetic Modelling of Saturated IPMSMs, for Improved Torque Estimation and Accurate MTPA Control

Master's thesis in Energy and Environmental Engineering
Supervisor: Robert Nilssen
Co-supervisor: Roy Nilsen, Aravinda Perera
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

NTNU
Norwegian University of
Science and Technology

Magnetic Modelling of Saturated IPMSMs, for Improved Torque Estimation and Accurate MTPA Control

Øyvind Sommer Klyve
Department of Electric Power Engineering
Norwegian University of Science and Technology
Supervisor: Robert Nilssen
Co-supervisors: Roy Nilsen, Aravinda Perera

***Abstract*—When aiming for optimal control of interior permanent magnet synchronous motors, a challenge is to develop accurate magnetic models describing the relation between the applied currents and the resulting flux linkages in the machine. A linear relation between the currents and the flux linkages is often assumed. However, because of magnetic saturation and cross-coupling phenomena, the motor magnetics are non-linear. In this thesis, to model the non-linear magnetic phenomena, several formulas differing in complexity and form are proposed and compared. The formulas are developed by coefficients curve fitted to calculated data from a FE model of a commercial motor developed in this thesis. At full load and when operating with maximum torque per ampere, the deviations from the FE computed q-axis flux linkage to the modeled one was 17.85%, when assuming linear magnetics. However, the deviation between the FE calculated and model estimated q-axis flux linkage was only 0.88% for the same operating point, when curve fitting one of the polynomial formulas to a wide range of FE measured operating points. When curve fitting the same polynomial formula to only 9 FE measurement points, the deviation only increased to 3.00% for the same operating point, proving the possibility of rapid development of accurate magnetic models for new undescribed motors during start-up.**

Using one of the developed polynomial magnetic models when calculating the current references to achieve maximum torque per ampere, the torque obtained at full load only increased by 0.04% compared to when using the linear magnetic model. However, for higher stator currents, the relative increase in torque got larger compared to the linear model. In addition, the deviation from the analytically calculated torque to the FE calculated torque was assessed. At full load, the deviation was 6.35% when assuming linear magnetics, but it was reduced to 2.89% when using a polynomial magnetic model. Finally, it was argued that the accuracy of the analytical torque equation could be improved by being extended to take into account the spatial harmonics of the magnetic flux density in the air gap.

The magnetic models presented and the related torque evaluations were to be verified at the lab during this thesis. However, the commissioning of the interior permanent magnet synchronous motor is still ongoing by the research group, such that the conclusions made from this thesis would have to be verified experimentally at a later stage.

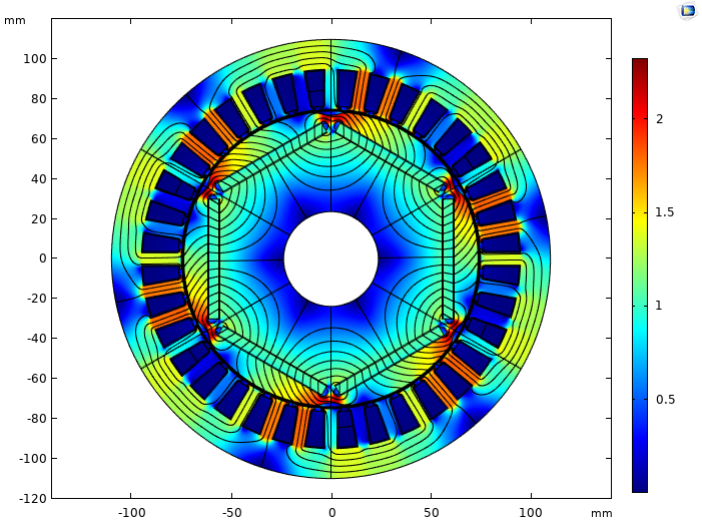


Fig. 1: Magnetic flux density in the FE model of the IPMSM at full load and pure q-axis current.

I. Introduction

Interior Permanent Magnet Synchronous Motors (IPMSM) are growing in the market due to their high torque density and efficiency [1]. Compared to surface-mounted Permanent Magnet Synchronous Machines (PMSM), the IPMSM can achieve higher rotational speeds, as the permanent magnets are protected and contained within the rotor iron [2]. However, the high reluctance of the permanent magnet material, located in the direct axis, will result in unequally distributed magnetic flux density in the rotor. This also results in a different non-linear magnetic behavior in both axes and a more complex magnetic cross-coupling between the direct (d) and quadrature (q) axes [3].

In order to achieve robust control ensuring accurate Maximum Torque Per Ampere (MTPA) and high performance in the field weakening region, the non-linear magnetic phenomena need to be taken into account [4]. In sensorless synchronous

motor drives, the control system also has to integrate the non-linearities in order to get the back-EMF observers to calculate the induce voltage accurately enough [5] [6]. Thus, a magnetic model of the motor, meaning the estimated relation between the flux linkages and the stator currents in the machine, has to be developed.

The flux linkages in the motor in relation to the applied currents can be analyzed using Finite Element (FE) modeling or by lab investigation. When mapping the relation by FE analysis [7] [8] [9], knowledge about the motor geometry and material properties is required. Experimentally, the flux linkages can be calculated when measuring the resistance, currents and induced back EMFs while either 1) accelerating and decelerating the motor spinning freely and calculate the flux linkages by the dynamic voltage equation [10], 2) rotating the motor at constant speed by another speed-controlled drive and calculate the flux linkages by the steady-state induced voltage equation [11] [12], 3) injecting voltages to the motor at standstill and calculating the flux linkages by integrating the applied voltage subtracted by the resistive voltage drop [13] [14].

When having mapped the flux linkages to the applied currents, a magnetic model representing the interconnection can be developed in the control scheme by either look-up tables [9] [15] or by defining explicit formulas based on the measured data [4] [8] [16] [17].

This thesis aims to study the motor magnetics by FE analysis using COMSOL Multiphysics, based on the geometry of a commercial IPMSM. The resulting flux linkages due to the applied stator currents in the FE model are mapped under operation conditions providing magnetic saturation and cross-coupling. In order to represent these phenomena in the control system of the electric drive, magnetic models with explicit formulas will be developed. The linear magnetic model is to be compared to formulas consisting of coefficients that are to be curve fitted by the linear least square method to the FE calculated data points. The formulas presented are combinations of pre-described formulas from the literature and empirically developed polynomial formulas. The performance of the explicit formulas is evaluated based on their simplicity, their fulfillment of the reciprocity condition, and their ability to model the magnetic saturation and cross-coupling in the motor accurately. The formulas' ability to predict the flux linkages outside of the original curve fitting area, will also be studied, as well as their curve fitting performance for few measurement points.

Based on the *linear* magnetic model and the *best* polynomial model, the MTPA trajectories in the regions of high saturation and cross-coupling will be developed. The proposed stator currents providing MTPA for both magnetic models will be simulated in COMSOL. The potential benefit of more torque per ampere when deriving the MTPA trajectories using the best polynomial formulas compared to the linear magnetic model is to be demonstrated. Finally, the torque calculated using the analytical torque equation will be compared to the FE calculated torque using Arkkio's method.

The thesis will thus give an extended understanding of how afflicted the IPMSM is to magnetic saturation and cross-coupling and how these phenomena can be modeled in the control system of the drive. In addition, the thesis is expected to demonstrate how the inclusion of detailed magnetic models is negligible when aiming to improve the MTPA. Suggestions to further improve the motor control will be given as well, based on the analysis of the torque equation.

II. Theory

In this section, the rotor oriented two phase per unit model of the IPMSM will be presented, to clarify how the magnetic modeling can be included in the motor model and thus the control system itself. Furthermore, a short introduction to the most common motor control strategies will be presented, to highlight how the magnetic modeling might improve the control of the IPMSM.

A. Per unit motor model

1) Model description: The per unit model is used to convert the measured SI-units from COMSOL into per unit values which can be interpreted by the control system. An advantage using per unit models, is that it is simple to detect if the motor is in an overloaded operating condition or not. In addition, experience from one motor drive can be easily transferred to another, as the variables and parameters in per unit values differ vaguely from motor to motor. The IPMSM electrical equivalent per unit model is given in the rotor reference frame, also called d- and q-axis reference frame, and common assumptions for the model are: [18]

1) Magnetic saturation and cross-coupling in the machine can be neglected.
2) The windings generate a sinusoidal distributed B-field in the air gap, meaning spatial harmonics are neglected [15].
3) The windings are symmetrical and the physically distributed windings can be represented as concentrated windings, which in turn generates a sinusoidal distributed field in the air gap.
4) The resistances, inductances and magnet properties are temperature and frequency independent.

TABLE I: Base values of the per unit model of the IPMSM

Base parameter	Expression
U_{base}	$\hat{U}_n$
I_{base}	$\hat{I}_n$
Z_{base}	$\hat{U}_n/\hat{I}_n$
Ψ_{base}	$\hat{U}_n/(2\pi \cdot f_n)$
T_{base}	$3/2 \cdot p \cdot \hat{U}_n \cdot \hat{I}_n/(2\pi \cdot f_n)$
N_{base}	N_n

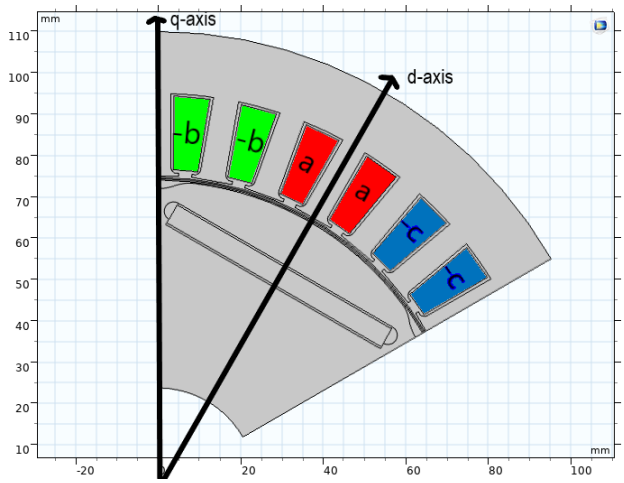


Fig. 2: Coil layout and axis definitions of the modelled sector of the motor.

The base values of the per unit model are given in Table I, where the base voltage, $\hat{U}_n$, and base current, $\hat{I}_n$, are the per phase amplitude values. Furthermore, f_n is the nominal electrical frequency, p the number of pair of poles and N_n the nominal rpm.

The d- and q-axis reference frame is defined based on the rotor saliency. The d-axis is aligned in the direction of the maximum reluctance, meaning the d-axis is fixed to the permanent magnet flux vector. Meaning a positive d-axis current enhances the magnetic flux along the permanent magnet, while a negative d-axis current opposes the the magnetic field of the permanent magnet. The q-axis is displaced by 90 electrical degrees relative to the d-axis, as seen in Figure 2.

Furthermore, the voltages and currents are Park transformed, such that the stator current vector $\underline{i}_s$ can either be expressed in either cartesian or polar coordinate systems, as given in Equation 1.

$$\underline{i}_s = i_d + j i_q = i_s \angle \theta \tag{1}$$

From complex number theory, Equation 2 and 3 are fulfilled.

$$i_s = \sqrt{i_d^2 + i_q^2} \tag{2}$$

$$i_d = \sqrt{i_s^2 - i_q^2} \tag{3a}$$

$$i_q = \sqrt{i_s^2 - i_d^2} \tag{3b}$$

From the defined load angle θ and stator current magnitude, i_s, Equation 4 is fulfilled as well. The stator current phasor is the reference phasor, such that it is aligned with the the positive d-axis when $\theta = 0$. The phasor rotates counterclockwise, such that when $\theta = \pi/2$, the stator current phasor is aligned with the positive q-axis.

$$i_d = i_s cos\theta \tag{4a}$$

$$i_q = i_s sin\theta \tag{4b}$$

The magnitude of the per unit stator voltage is defined as in Equation 5.

$$u_s = \sqrt{u_d^2 + u_q^2} \tag{5}$$

The equation of the induced voltages, u_d, u_q, in the per unit rotor reference frame as functions of the per unit currents, i_d, i_q, and per unit flux linkages, ψ_d, ψ_q, can be simplified to Equation 6 [18].

$$u_d = r_s i_d + \frac{1}{\omega_n}\frac{d\psi_d}{dt} - n\psi_q \tag{6a}$$

$$u_q = r_s i_q + \frac{1}{\omega_n}\frac{d\psi_q}{dt} + n\psi_d \tag{6b}$$

Where r_s is the stator resistance in per unit, ω_n the nominal angular frequency, and n the per unit value of the rpm of the motor. By the same per unit system, the induced electric torque can be simplified to Equation 7, when assuming sinusoidally distributed magnetic flux density in the air gap [18].

$$\tau_e = \psi_d i_q - \psi_q i_d \tag{7}$$

The resistance, voltages, currents and angular speed can be measured on the outside of the physical motor. However, the flux linkages have to be modelled in order for the analytical formulas for the torque and voltages to be accurate. The d- and q-axis flux linkages are dependent on the applied d- and q-axis stator currents.

2) The linear model: As mentioned, a common assumption for the per unit model is that the magnetic saturation and cross-coupling can be neglected, meaning there are assumed linear relationships between the respective d- and q-axis flux linkages to the d- and q-axis currents. This way of modeling the motor magnetics will be referred to as the linear magnetic model, and the model is given in Equation 8.

$$\psi_d = x_d i_d + \psi_m \tag{8a}$$

$$\psi_q = x_q i_q \tag{8b}$$

Where x_d and x_q are the constant per unit reactances due to the d- and q-axis inductances respectively. In addition, ψ_m is the per unit flux linkage due to the permanent magnet, as it will produce a constant flux linkage along the d-axis independent of the applied currents. As the linear magnetic model does not take the non-linear magnetic saturation and cross-coupling phenomena into account, other ways to model the motor flux linkages in relation to the applied stator currents will be presented.

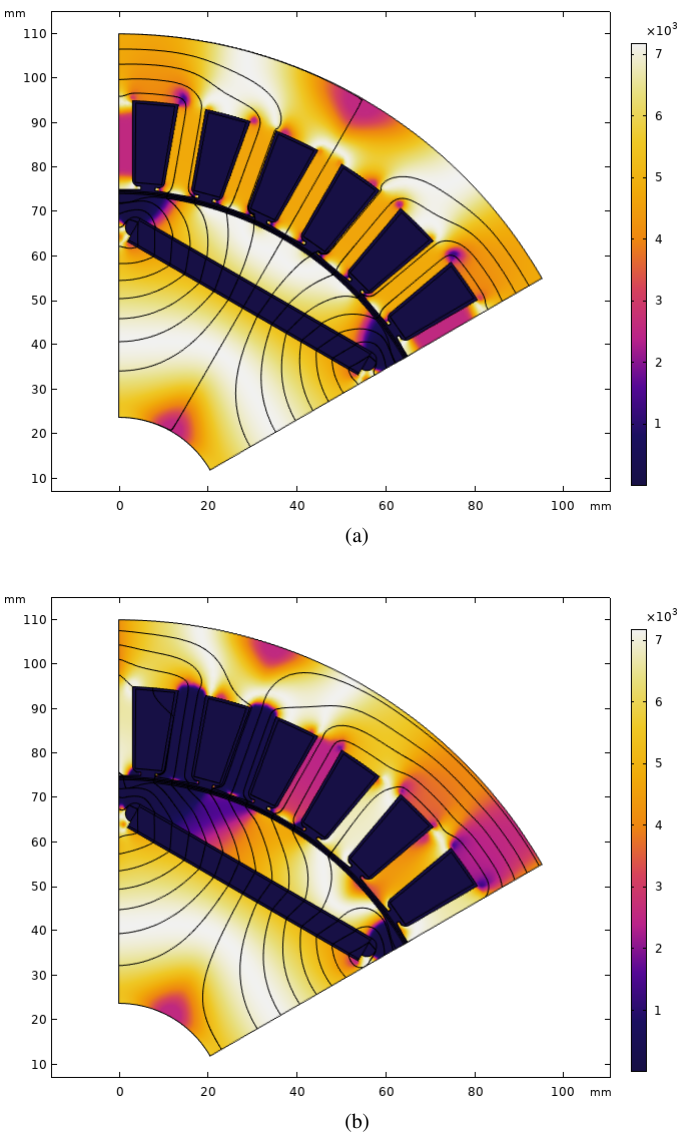
B. Magnetic Modelling

For IPMSMs under heavy loading conditions, there will be magnetic saturation in the ferromagnetic stator and rotor material and cross-coupling between the d- and q-axis. Cross-coupling is a phenomenon that for instance occurs when q-axis currents produce flux in the q-axis, saturating the iron along the d-axis as well, as visualised in Figure 3. When the reluctance along the d-axis path increases, the d-axis flux linkage will be reduced even though there were no d-axis currents applied. The same phenomenon occurs for the q-axis flux linkage for applied d-axis currents. Meaning the magnetic flux produced by one axis current might distort the flux produced by the other axis current [3]. Thus, if the motor control system has to operate accurately and efficiently, the linear model might be too simple, and instead each flux linkage has to be modelled to be dependent on both the d- and q-axis current, as $\psi_d(i_d, i_q)$ and $\psi_q(i_d, i_q)$.

When modelling the flux linkages it is assumed that the q-axis flux linkage will be more affected by magnetic saturation compared to the d-axis flux linkage. This is due to the fact that the permanent magnet ensures a high magnetic flux density in the rotor iron above the permanent magnet and in the stator teeth. When applying positive q-axis currents, the magnetic flux density will increase in these areas saturating the motor further. As will be explained in Section II-C, optimal torque control is achieved by applying negative d-axis currents. This means that when d-axis currents are applied in the motor, they will be of negative value setting up magnetic flux opposing the flux from the permanent magnet. Thus, the total magnetic flux density in the motor is reduced, and the saturation would be reduced as well. This led to the assumption that the q-axis flux linkage is thought to be more influenced by magnetic saturation than the d-axis flux linkage.

The non-linear relation between the flux linkages and the applied currents due to saturation and cross-coupling, can be included in the control system of the electrical drive by either constructing look-up tables or defining explicit formulas.

1) Look-up Tables: The look-up tables in the control structure can be modeled such that for given i_d, i_q currents, the table will export either ψ_d or ψ_q flux linkages. The look-up tables can also be defined with the flux linkages as input variables, exporting the currents, in those parts of the control structure where this is practical. In order to build the look-up tables, the motor must either be modelled and measured using FE software or be physically measured for a wide range of operating points, to find the resulting d- and q-axis flux linkages for the applied d- and q-axis currents. As it is impossible to measure all possible operating points, the control software will have interpolate between the measured



(a)

(b)

Fig. 3: The relative permeability in the motor, representing the saturation, at (a) no-load and (b) full load of pure q-axis current. From (b) it can be seen how q-axis current saturates parts of the stator teeth and rotor iron. Due to the saturated iron, the reluctance of the d-axis flux path will increase, such that the flux linkage along the d-axis will be reduced, even though no negative d-axis current has been applied. This phenomenon is called cross-coupling.

data in the table during operation [4]. However, if the motor in a transient state operates outside of the measured table data, the control structure might become unstable.

2) Explicit Functions: By defining models having explicit formulas for the relation between the d- and q-axis flux linkages and currents, the control system may operate unbounded by a predefined range as it may extrapolate outside the measured data range. In addition, the related controller may require less data storage compared to the look-up tables [19] [20]. The models might include parameters and exponents which have to be curve fitted to measured values, often by the linear least square method [13]. If a look-up table has been developed, the formulas of the models can easily be curve fitted to the table data.

a) The polynomial models: Polynomial models can be defined based on Equation 9, but with selecting some coefficients to be curve fitted and some to be put to zero to simplify the equation. Equation 9 is given with the parameter ψ_{dm} which will be explained in Section III. Several polynomial models have been presented and show good results [8] [16] [17].

$$\psi_{dm} = \sum_{i=0}^{n}\sum_{j=0}^{m} d_{i,j} i_d^i i_q^j \tag{9a}$$

$$\psi_q = \sum_{i=0}^{n}\sum_{j=0}^{m} q_{i,j} i_d^i i_q^j \tag{9b}$$

The polynomial models often require several FE data points to be curve fitted to and seldomly take the reciprocity condition, given in Equation 10, into account. By fulfilling the reciprocity condition, the flux linkages will be modelled such that their corresponding inductances will not be able to generate or dissipate energy [21].

$$\frac{\partial i_d}{\partial \psi_q} = \frac{\partial i_q}{\partial \psi_d} \tag{10}$$

Depending on the control structure, the flux linkages can be expressed as functions of the currents, or the currents as functions of the flux linkages. For the formulas proposed in this thesis, any i_d, i_q currents can be solved numerically from the functions of ψ_d, ψ_q, and opposite.

b) The exponential model: One of the proposed models taking the reciprocity condition into account is given in Equation 11 [4]. The *exponential* model is also defined to solve the d- and q-axis currents based on known d- and q-axis flux linkages.

$$i_d = (a_{d0} + a_{dd}|\psi_d|^{\alpha} + \frac{a_{dq}}{\delta+2}|\psi_d|^{\gamma}|\psi_q|^{\delta+2})\psi_d - i_f \tag{11a}$$

$$i_q = (a_{q0} + a_{qq}|\psi_q|^{\beta} + \frac{a_{dq}}{\gamma+2}|\psi_d|^{\gamma+2}|\psi_q|^{\delta})\psi_q \tag{11b}$$

Where i_f models the MMF due to the permanent magnets, and a_{d0}, a_{dd}, a_{q0}, a_{qq} and a_{dq} are positive coefficients and α, β, γ and δ positive exponents. Equation 11 will be referred to as the *exponential* model in this thesis, and its performance to model the motor magnetics will be compared to the performance of the polynomial models.

C. Control strategy

As mentioned, as the torque equation of the equivalent electrical model of the IPMSM is dependent on the flux linkages, the estimated electromagnetic torque produced by the motor might be more accurate when the magnetic model is improved by including saturation and cross-coupling. In order to operate the motor as efficiently as possible, it should be operated using Maximum Torque Per Ampere (MTPA) control. To find the operating points to achieve MTPA, combinations of the d- and q-axis currents are deduced from the torque equation to ensure as high torque as possible for any total stator current, i_s. In addition, the control system estimates the produced torque in the motor based on the d- and q-axis currents fed to the machine. Thus, in both cases, if the magnetic model is inaccurate, the control system will not provide the correct current references to achieve MTPA and the produced torque might be estimated incorrectly by the control system.

1) Torque production: MTPA will be achieved by applying combinations of negative d-axis and positive q-axis currents. This can be easily demonstrated by assuming linear magnetics and combining Equation 7 and 8, into Equation 12.

$$\tau_e = \psi_m i_q - (x_q - x_d) i_d i_q \tag{12}$$

As the d-axis was defined along the axis with maximum reluctance, the per unit q-axis reactance will be greater than the d-axis per unit reactance, $x_q > x_d$. Thus, the torque can be maximized with negative d-axis and positive q-axis currents. This is also the case when the motor magnetics are modelled to be non-linear. Figure 4 shows how the maximum torque is achieved for the FE modeled motor when applying negative d-axis and positive q-axis current at full load, meaning for a total stator current of 1pu.

The maximum torque is achieved for negative i_d and positive i_q currents due to the permanent magnet field-alignment torque and the reluctance torque. The first term in Equation

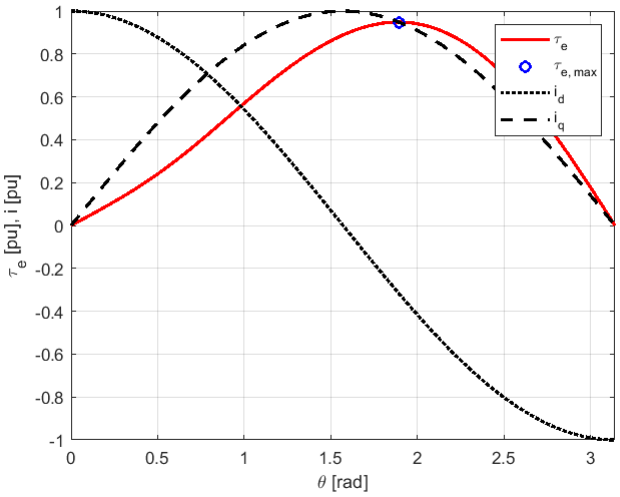


Fig. 4: The maximum torque for a total stator current of 1pu is achieved by a loading angle between $\pi/2$ and π, meaning a positive i_q and negative i_d current.

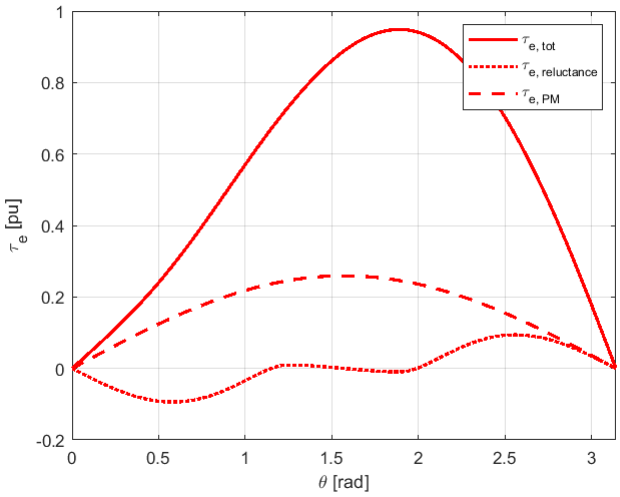


Fig. 5: Torque for different load angles for 1pu stator current, showing how the torque curve of the IPMSM can be interpreted as the sum of the permanent magnet and reluctance torque curves. The permanent magnet torque curve was developed by setting the relative permeability of the rotor iron to 1, while the reluctance torque curve was developed by setting the remanence flux of the magnet to 0 in the FE model.

12 represents the field-alignment torque produced as a result of the interaction of the d-axis permanent magnet flux to the flux produced by the q-axis stator current, as for conventional surface mounted PMSMs. The second term represents the reluctance torque, produced due to the saliency in the rotor [2]. This is visualized in Figure 5, where $\tau_{e,reluctance}$ is the torque achieved in the COMSOL model of the IPMSM when the remanence flux density of the permanent magnet is set to zero, such that it is only the reluctance torque which is present. Positive torque is achieved in the area where $\theta \in [\pi/2, \pi]$, meaning positive i_q current and negative i_d current. From the same figure, the $\tau_{e,PM}$ shows how the field alignment torque is at its maximum for pure q-axis current. The field alignment torque curve was achieved by setting the rotor iron relative permeability to 1 in the COMSOL model. The total torque $\tau_{e,tot}$ for different load angle can thus be interpreted as the combination of these two torque producing phenomena, such that the maximum torque is achieved for negative d-axis current and positive q-axis current.

2) MTPA: The operating point where the motor is inducing a given torque for as little stator current as possible, is called the MTPA operating point. The maximum achievable torque for each total stator current, i_s, can explicitly be found by finding the extremal point of the torque equation in Equation 7, with respect to either the d- or q-axis current. It is arbitrary whether the extremal point is found by the d- or q-axis current, as they can be written as functions of each other and the defined total stator current, as given in Equation 3. Meaning if the derivative of the torque with respect to the d-axis current is to be found, the q-axis current can be expressed as a function of the chosen total stator current and the d-axis stator current, such that the torque equation has only the d-axis current as an unknown. Thus, the operating point for the selected total stator current can be found by Equation 13, when defining the torque by either i_d or i_q.

$$\frac{\partial \tau_e}{\partial i_d} = 0 \tag{13a}$$

$$\frac{\partial \tau_e}{\partial i_q} = 0 \tag{13b}$$

Having calculated either i_d ensuring MTPA for a given i_s, i_q can be found by Equation 3, and vice versa.

3) Field Weakening: When accelerating the motor up to its base speed, the preferred control strategy is the MTPA strategy. However, for the motor to accelerate beyond this point, the flux in the machine has to be reduced, to ensure the induced voltage in the stator windings is not greater than the maximum inverter voltage [22]. Reducing the magnetic flux in the motor can be achieved by applying additional negative i_d current, producing magnetic flux opposing the flux set up by the permanent magnet, such that the total flux linkage in the motor is reduced. The different control strategies during field weakening operation will not be covered in this thesis, but it should be noted that the motor will always operate with positive i_q and negative i_d current in both MTPA and field weakening operation, to generate a positive torque for positive speed.

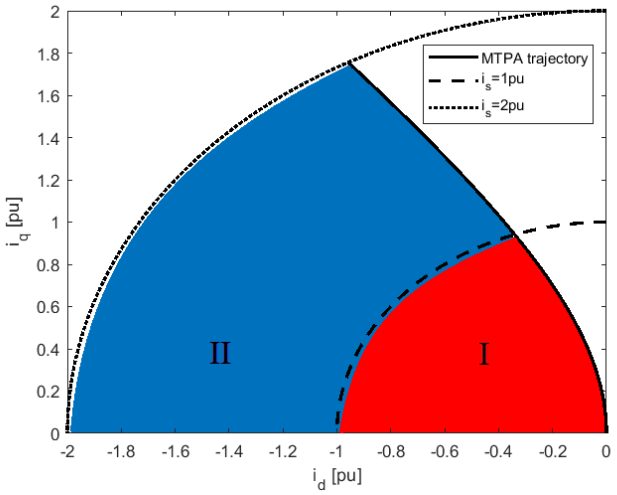


Fig. 6: Operating regions of the motor during Maximum Torque Per Ampere (MTPA) and Field Weakening (FW) control strategy for I) maximum 1pu and II) maximum 2pu stator current. The MTPA trajectory was developed assuming linear magnetics, meaning constant x_d, x_q. The MTPA curve represents the best combination of i_d and i_q currents for any stator current i_s, to obtain maximum torque.

In both MTPA and field weakening operation, the i_d, i_q currents are bound by the stator current limit, $i_{s,max}$. In Figure 6 the curves of $i_s = 1pu$ and $i_s = 2pu$ are drawn together with the trajectory of combinations of d- and q-axis currents providing MTPA when using the linear magnetic model. As mentioned, at base speed, the motor should be operated using the MTPA strategy, but to further accelerate the motor it has to go into field weakening by increasing negative i_d currents. Thus, area I in Figure 6 represents all operating areas for all total stator currents less than 1pu. Area II represents all possible operating areas if the motor is allowed to operate up to total stator currents of 2pu, for instance in a transient state. The MTPA trajectory and maximum stator current curves will be of interest when the measured flux linkages in relation to the applied currents are to be compared to the formula modelled flux linkages. The goal of the magnetic modeling is that the difference between the curve fitted models and the FE calculated flux linkages should be small within these operating areas.

4) Control system implementation: During this thesis, the magnetic modeling was not implemented in a control system simulation. However, how the control system is built is relevant knowledge when analyzing how the magnetic modeling of the motor could improve the control of the motor. Figure 7, shows the control system for a current controlled drive in rotor coordinates with a position sensor. Before operation, the motor magnetic model should be identified, either by FE analysis or measurements at the lab.

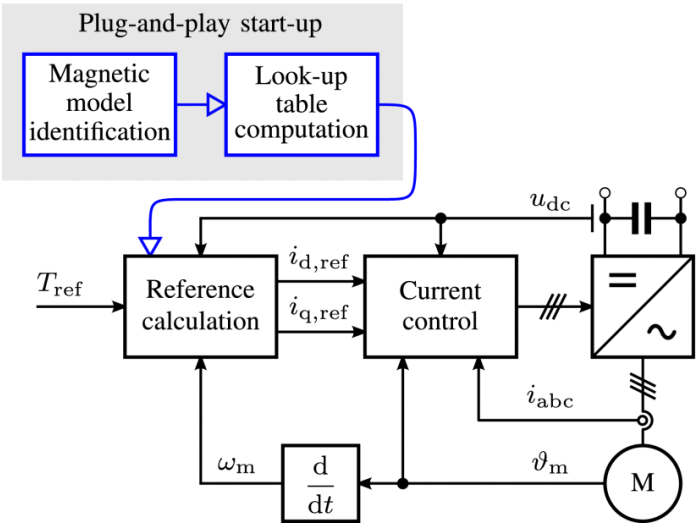


Fig. 7: Simplified structure of the control system for an IPMSM, showing how the magnetic model is integrated in the current reference calculation, to achieve optimal control. The figure has been taken from [4].

Then either look-up tables or explicit functions should be defined and incorporated in the current reference calculation. The current references are calculated based on the angular speed of the motor and the torque reference. If the magnetic model is accurate, the control system would benefit in two ways. Firstly, the proper current references ensuring MTPA would be correctly derived, providing efficient operation of the motor. Secondly, the control system would be able to correctly estimate if the motor is achieving the same torque as given by the torque reference, only based on the measured input currents and the magnetic models. Based on the current references, the current controller then provides the corresponding gate signals to the inverter to achieve MTPA. The motor can be speed controlled, by connecting a speed controller to the torque reference. By this measure, the speed controller could ensure that the torque reference is adjusted such that the load will be fed its required torque, even though the torque estimation in the control system is inaccurate. This has been proven in Section VI to be advantageous, as the analytically calculated torques differed from the COMSOL simulated ones, due to the simplified torque equation.

III. Method: Magnetic Modeling

During this section, the approach to calculate the d- and q-axis flux linkages in relation to applied d- and q-axis currents in the developed FE model is to be described. Furthermore, the method to curve fit the proposed models to the FE calculated data is shown, as well as how the corresponding surface plots to the defined models can be plotted using MATLAB. The results can be found in Section V.

A. FE modeling and data collection

1) Building the model: The COMSOL model of the IPMSM was based on a physical motor, a WEG W22, which was opened and measured. The stator and rotor geometry parameters can be found in Appendix A. The stator windings in the motor were distributed, bifurcated, single layer, integer windings with coil ends of 3 levels, as seen in Figure 8. The material and winding properties were not given and had to be estimated. The COMSOL model was built assuming a star-connected motor, even though the WEG W22 was delta-coupled, which had to be taken into account when building the model. The turn number was assumed to be 102 such that the induced phase voltage at no-load conditions in the star-coupled COMSOL model resulted in the given line-to-line voltage at the nameplate of the physical motor. The base value of the current was set to be $1/\sqrt{3}$ of what was given at the nameplate to compensate for the delta-to-star change done in the COMSOL model. For delta-coupled machines, circulating third harmonic currents might be present and would thus not be included in the star-coupled COMSOL model. However, as the turn number and total currents in the stator windings were assumed to be close to that of the physical motor, the trends of the observed magnetic saturation and cross-coupling in the COMSOL model should be valid for the physical motor as well.

The electrical steel in the model was chosen to be a typical, commercially available material, *Silicon Steel NGO 35PN300*, and the permanent magnet was assumed to be of the material *N50 (Sintered NdFeB)*. The coil material was set to *Copper*, and the steel sheets between the permanent magnet and the rotor were set to be of *Steel AISI 4340*. All other domains were set to be *Air*. A sector of the motor, one sixth, was modeled, meaning the measured motor length was multiplied by 6 in the FE model to correctly calculate the induced torque, voltages and flux linkages, despite having modeled only a sector. The winding scheme can be found in Figure 2 and the remanence flux direction of the permanent magnet was set radially towards the stator, along the d-axis.

The COMSOL model was current-controlled, meaning the three-phase currents in the coils were defined as given in Equation 14. The capital letters emphasize that the currents are given in SI units, as required by COMSOL. Where I_s is the amplitude value of the three-phase currents.

$$I_a = I_s sin(\omega t - \theta) \tag{14a}$$

$$I_b = I_s sin(\omega t - \frac{2\pi}{3} - \theta) \tag{14b}$$

$$I_c = I_s sin(\omega t + \frac{2\pi}{3} - \theta) \tag{14c}$$

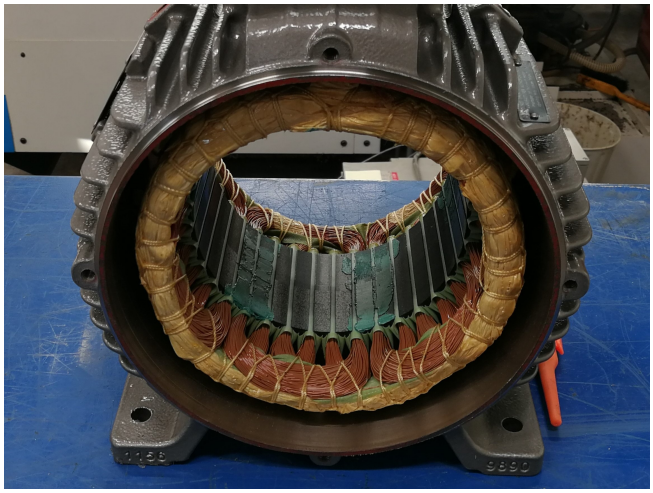


Fig. 8: Stator of the physical motor which the FE model was based on.

When solving the model, the rotor position was held constant while the position of the current vectors in relation to the rotor axis were swept using *Stationary*, *Parametric Sweeps*. As the rotor position was constant in relation to the stator, the torque and indcutance, ripples were excluded from the study. As the study was stationary, t was equal to zero. However, the ωt term was included in the model such that dynamic analysis would be possible.

2) Park Transform: The park transform was used to convert the three-phase sinusoidal currents into a two-phase rotor-oriented reference frame with constant d- and q-axis currents for a given I_s and θ. For the defined currents formulas and axis-definitions, the I_d and I_q currents, corresponding to the applied three-phase currents I_a, I_b and I_c, were calculated as given in Equation 15, based on [3]. The $-\pi/2$ term was added to correctly align the induced flux due to I_d currents with the defined d-axis through the permanent magnet.

$$I_d = \frac{2}{3}(I_a cos(\omega t - \frac{\pi}{2}) + I_b cos(\omega t - \frac{2\pi}{3} - \frac{\pi}{2}) + I_c cos(\omega t + \frac{2\pi}{3} - \frac{\pi}{2})) \tag{15a}$$

$$I_q = \frac{2}{3}(I_a sin(\omega t - \frac{\pi}{2}) + I_b sin(\omega t - \frac{2\pi}{3} - \frac{\pi}{2}) + I_c sin(\omega t + \frac{2\pi}{3} - \frac{\pi}{2})) \tag{15b}$$

As defined in Equation 4, a current stator angle, θ, of $0 rad$ corresponds to a pure I_d current enhancing the magnetic flux in the direction of the flux set up by the permanent magnet, and $\pi/2\ rad$ corresponds to a pure I_q current bending the flux lines from the permanent magnet and creating torque. The three-phase currents were directly calculated by the physics package *Rotating Machinery, Magnetic*, such that for instance

the I_a current was found by the expression *rmm.ICoil_a*.

COMSOL also calculated the flux linkage by integrating the magnetic flux density through each of the three coils of phase a, b and c. The expression for instance for the flux linkage in coil a is given as *rmm.PhiCoil_a* in COMSOL, and is called the coil concatenated flux. The corresponding d- and q-axis flux linkages were calculated using the Park transform in the same manner as the currents, as given in Equation 16.

$$\Psi_d = \frac{2}{3}(\Psi_a cos(\omega t - \frac{\pi}{2}) + \Psi_b cos(\omega t - \frac{2\pi}{3} - \frac{\pi}{2}) + \Psi_c cos(\omega t + \frac{2\pi}{3} - \frac{\pi}{2})) \tag{16a}$$

$$\Psi_q = \frac{2}{3}(\Psi_a sin(\omega t - \frac{\pi}{2}) + \Psi_b sin(\omega t - \frac{2\pi}{3} - \frac{\pi}{2}) + \Psi_c sin(\omega t + \frac{2\pi}{3} - \frac{\pi}{2})) \tag{16b}$$

3) Inverse Park Transform: As the control strategy is based on the per-unit motor model in the two-phase rotor-oriented reference frame, it is of interest to find the a, b, c currents that are needed to achieve preselected I_d, I_q currents. An inverse Park transform was developed to calculate the a, b, c currents needed to obtain the desired I_d, I_q currents, such that the corresponding Ψ_d, Ψ_q flux linkages could be calculated.

In order to map Ψ_d and Ψ_q in relation to I_d and I_q, the amplitude I_s and the load angle θ had to be calculated by the preselected I_d and I_q, as given in Equation 17 and 18. By inserting the calculated I_s, θ, into the equations for the stator currents in Equation 14, the desired I_d, I_q currents were achieved.

$$I_s = \sqrt{I_d^2 + I_q^2} \tag{17}$$

$$\theta = atan2(\frac{I_q}{I_d}) \tag{18}$$

The implementation of the method is visualized in Figure 10 and Figure 11, showing the magnetic flux in the motor without applied currents due to the permanent magnet, also how a positive d-axis current enhances the flux along the d-axis, as well as how a negative d-axis current opposes the flux from the permanent magnet. Applying pure q-axis currents bends the flux lines of the permanent magnet creating torque on the rotor. Table II shows the calculated d- and q-axis flux linkages for the applied predefined d- and q-axis currents, confirming the assumptions made of the flux in the machine based on Figure 10 and Figure 11.

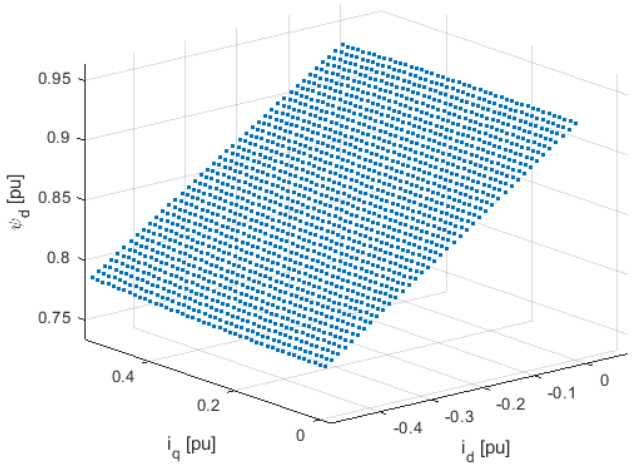


Fig. 9: Measurements points from COMSOL of the flux linkages for different combinations of i_d and i_q currents.

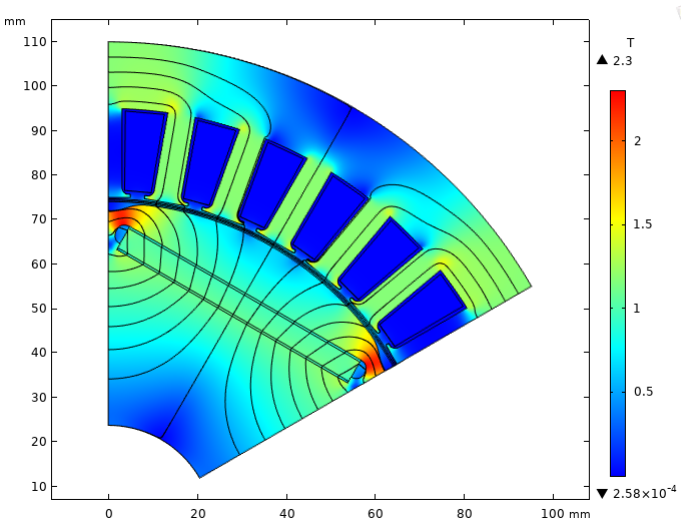
TABLE II: Flux linkages in relation to applied currents calculated by the model. For negligible applied currents, there is a constant flux linkage in the d-axis, Ψ_m, due to the permanent magnet. When applying positive d-axis current, the d-axis flux linkage increases while negative d-axis current decreases the d-axis flux linkage. For positive q-axis currents, the q-axis flux linkage increases, while the d-axis flux linkage gets slightly reduced from Ψ_m, due to the cross-coupling as visualized in Figure 3.

$I_d[A]$	$I_q[A]$	$\Psi_d[Wb]$	$\Psi_q[Wb]$
$-1e-9$	$1e-9$	1.6781	$5.6295e-5$
$\sqrt{2}\cdot 2.85$	$1e-9$	2.1755	$-2.0840e-4$
$-\sqrt{2}\cdot 2.85$	$1e-9$	1.1095	$-4.4033e-5$
$-1e-9$	$\sqrt{2}\cdot 2.85$	1.6629	1.0938

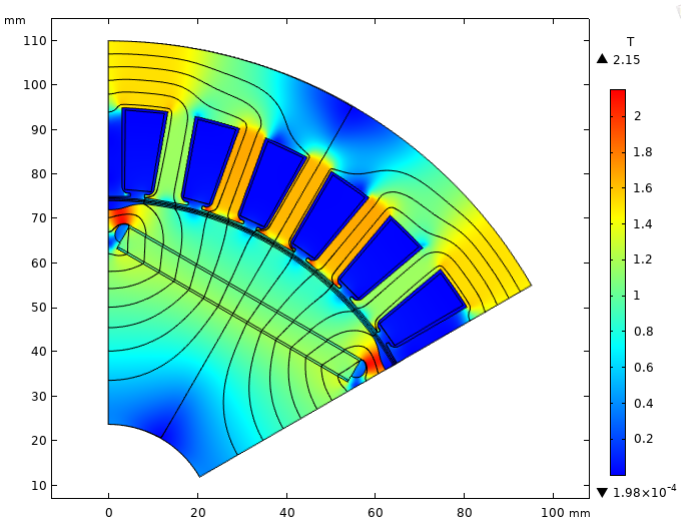
4) Mapping the flux linkages: The flux linkages were mapped for d-axis currents from $-1e-6A$ to $-2\cdot\sqrt{2}\cdot 2.85A$ with steps of -0.05, and q-axis currents from $1e-6A$ to $2\cdot\sqrt{2}\cdot 2.85A$ with steps of 0.05. Neither I_d nor I_q could be equal to zero, due to the calculation of the inductances where the currents are in the denominator. As given in Table III, the stator current base value was chosen to be $\sqrt{2}\cdot 2.85A$, meaning the mapping was not just done up to full load at 1pu, but also up to 2pu to model transient operation conditions as well. Figure 9 visualizes some of the total 26244 individual data points within the measured region. From *Derived Values* in COMSOL, the I_d, I_q, Ψ_d, Ψ_d vectors were exported as .csv files, to be analysed in MATLAB.

B. MATLAB post processing

In MATLAB, the FE calculated motor data had to be represented in the defined per-unit system.

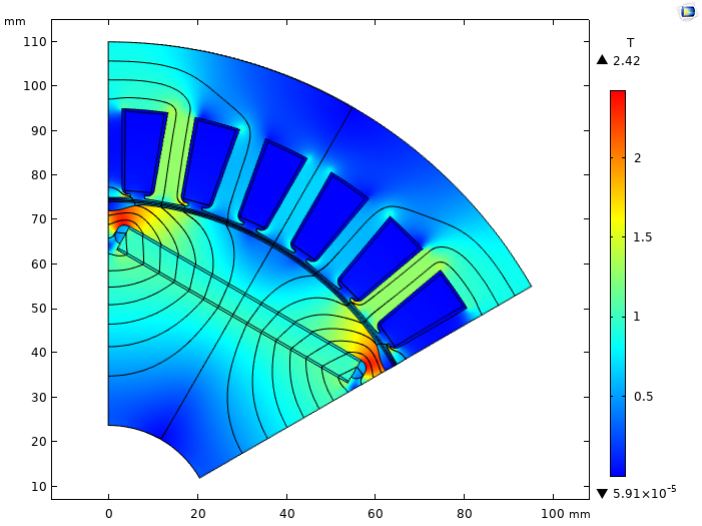


(a) $I_d = 1e-9A$, $I_q = 1e-9A$

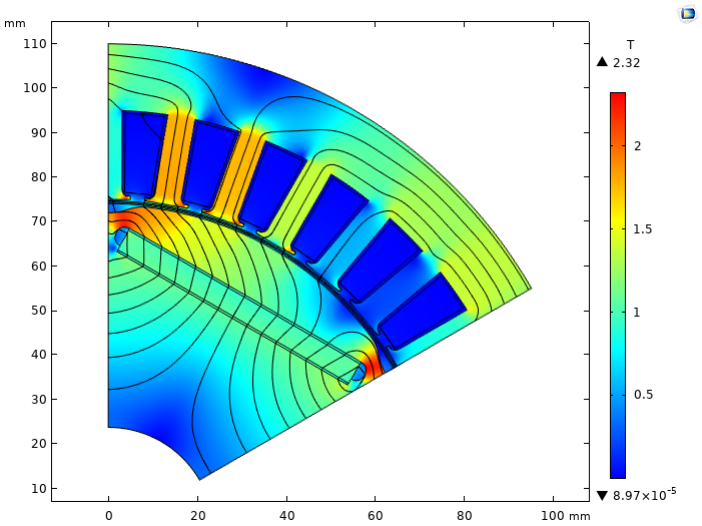


(b) $I_d = \sqrt{2} \cdot 2.85A$, $I_q = 1e-9A$

Fig. 10: Plot of normalized magnetic flux density and flux lines for the sector of the motor model in COMSOL for (a) no stator currents and (b) positive d-axis current. Showing how positive d-axis current is enhancing the flux from the permanent magnet.



(a) $I_d = -\sqrt{2} \cdot 2.85A$, $I_q = 1e-9A$



(b) $I_d = 1e-9A$, $I_q = \sqrt{2} \cdot 2.85A$

Fig. 11: Plot of normalized magnetic flux density and flux lines for the sector of the motor model in COMSOL for (a) negative d-axis current and (b) positive q-axis current. Showing how a negative d-axis current opposes the flux of the permanent magnet while a positive q-axis current bends the flux lines of the permanent magnet counterclockwise, providing torque.

1) Numerical values of the per-unit model: The chosen numerical values for the per-unit model, presented in Section II-A, are based on the nameplate of the modeled physical WEG motor. The nominal amplitude phase current, $\hat{I}_n$, after the delta-to-star transformation was set to $\sqrt{2} \cdot 2.85A$, while the nominal amplitude phase voltage, $\hat{U}_n$, was set to $\sqrt{2} \cdot 400V$. The motor has 3 pair of poles, p, and nominal speed, N_n, of $1000rpm$ which corresponds to an electrical frequency, f_n, of $50Hz$, as summarized in Table III. Variables given in per-unit values are written with small letters, and vectors of values are given in a bold font.

C. Magnetic modeling

In this section, the magnetic models of the motor will be developed, meaning the formulas calculating the flux linkages as functions of the currents, or vice versa. Five magnetic models is to be compared, 1) the *linear* model, 2) the *simple* polynomial model, 3) the *best* polynomial model, 4) the *reciprocal* polynomial model and 5) the *exponential* model. It should be noticed that for all the proposed models, it is

TABLE III: Chosen numerical values for the base value calculations for the per-unit model

Expression	Value
$\hat{U}_n$	$\sqrt{2} \cdot 400[V]$
$\hat{I}_n$	$\sqrt{2} \cdot 2.85[A]$
f_n	$50[Hz]$
N_n	$1000[rpm]$
p	3

possible to define corresponding per-unit reactances by the formula given in Equation 19.

$$x_d(i_d, i_q) = \frac{\psi_d(i_d, i_q) - \psi_m}{i_d} \tag{19a}$$

$$x_q(i_d, i_q) = \frac{\psi_q(i_d, i_q)}{i_q} \tag{19b}$$

However, for simplicity, this thesis aims to model either the d- and q-axis flux linkages directly from the d- and q-axis currents, or opposite, without defining any reactances. The only model defining per-unit reactances is the *linear* magnetic model.

1) The linear model: As explained in Section II-A, the magnetic model of the IPMSM is often simplified by assuming linear magnetics, meaning that x_d and x_q are constant. Thus, magnetic saturation and cross-coupling has been neglected. The formulas are given in Equation 20.

$$x_d = \frac{\psi_d - \psi_m}{i_d} \tag{20a}$$

$$x_q = \frac{\psi_q}{i_q} \tag{20b}$$

In order to calculate the unsaturated x_d and x_q per-unit reactances, ψ_d and ψ_q were measured in the COMSOL model when applying i_d and i_q currents corresponding to $0.1pu$ respectively [3]. In addition, to calculate x_d, the constant flux linkage due to the permanent magnet, ψ_m, had to be calculated. In SI-units, Ψ_m was found to be $1.6781Wb$ as given in Table II, when I_d and I_q were close to zero. The calculated per-unit reactances x_d, x_q can be found in Table IV.

2) Curve fitting: For the polynomial and exponential models, the unknown coefficients and exponents of the functions had to be estimated by applying the linear least square method to the FE calculated data. A new parameter, ψ_{dm}, was defined as given in Equation 21, such that produced flux linkages due to the applied currents were independent of the flux linkage of the permanent magnet.

$$\psi_{dm} = \psi_d - \psi_m \tag{21}$$

In order to calculate the unknown coefficients, the formulas had to be linearized on the form given in Equation 22. $\mathbf{B}$ is the vector containing the unknown coefficients, which had to be solved by the linearized least square method by the values in the known vectors $\mathbf{Y}$ and matrix $\mathbf{X}$. How these vectors and matrices are built for each proposed formula is to be presented for each curve fitting model.

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} \tag{22}$$

Having linearized the formulas on the form given in Equation 22, the linear least square method could be solved using Equation 23 [23], incorporated in the MATLAB function *regress*.

$$\mathbf{B} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{23}$$

3) The polynomial models: As it had been observed a polynomial behavior in the plots of the flux linkages in relation to the currents, some of the magnetic models were built using polynomial functions. The proposed formulas were defined based on Equation 9, with the goal of having an accurate curve fit, but with as few unknown coefficients as possible.

For all linearized formulas, it should be noted that when the current vectors are multiplied or squared, they are done so element wise and not by matrix multiplication. Vector $\mathbf{0}$ has the same length as the current- or flux linkage vectors but contains only zeros.

a) The simple polynomial model: Based on the plots of the measured flux linkages in relation to the applied currents given in Section V, the d-axis flux linkage was observed to be quite linear in relation to the d-axis current, while the q-axis flux linkage had a second-order behavior in relation to the q-axis current. Thus, the *simple* polynomial model was defined, as given in Equation 24.

$$\psi_{dm,s} = d_{10s} i_d \tag{24a}$$

$$\psi_{q,s} = q_{01s} i_q + q_{02s} i_q^2 \tag{24b}$$

It should be noted that d_{10s} will not equal x_d from Equation 20, as x_d was calculated based on measured data from one operating point. The per-unit reactance, d_{10s}, was however curve fitted to give the smallest error in the whole defined operating range of the motor.

For this magnetic model, there are only three unknown parameters, and the reciprocity condition is fulfilled. However, saturation has been neglected for the d-axis flux

linkage, while the cross-coupling has been neglected for both the d- and q-axis flux linkage.

The simple polynomial formula can be linearized on the form given in Equation 22 into Equation 25.

$$\boldsymbol{Y_s} = \begin{bmatrix} \boldsymbol{\psi_{dm,s}} \\ \boldsymbol{\psi_{q,s}} \end{bmatrix} \tag{25a}$$

$$\boldsymbol{X_s} = \begin{bmatrix} \boldsymbol{i_d} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{i_q} & \boldsymbol{i_q^2} \end{bmatrix} \tag{25b}$$

$$\boldsymbol{B_s} = \begin{bmatrix} d_{10s} & q_{01s} & q_{02s} \end{bmatrix}^T \tag{25c}$$

b) The best polynomial model: By trial and error, the formulas fitting the measured data the best while having a small number of unknown coefficients were the formulas given in Equation 26. When modeling the cross-coupling, it was found beneficial not to have common coefficients for the d- and q-axis flux linkage as the sum of residuals would increase. In addition, the reciprocity condition is not fulfilled. Thus, the proposed formulas do not ensure a conservative system. They must therefore be considered as an empirical representation of the measured motor magnetics and not an attempt to develop a universal formula true to nature.

$$\psi_{dm,b} = d_{10b} i_d + d_{11b} i_d i_q + d_{02b} i_q^2 \tag{26a}$$

$$\psi_{q,b} = q_{01b} i_q + q_{02b} i_q^2 + q_{12b} i_d i_q^2 + q_{20b} i_d^2 \tag{26b}$$

The unknown coefficients of the model were found by solving the linearized system of equations, given in Equation 27, using the linear least square method.

$$\boldsymbol{Y_b} = \begin{bmatrix} \boldsymbol{\psi_{dm,b}} \\ \boldsymbol{\psi_{q,b}} \end{bmatrix} \tag{27a}$$

$$\boldsymbol{X_b} = \begin{bmatrix} \boldsymbol{i_d} & \boldsymbol{i_d i_q} & \boldsymbol{i_q^2} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{i_q} & \boldsymbol{i_q^2} & \boldsymbol{i_d i_q^2} & \boldsymbol{i_d^2} \end{bmatrix} \tag{27b}$$

$$\boldsymbol{B_b} = \begin{bmatrix} d_{10b} & d_{11b} & d_{02b} & q_{01b} & q_{02b} & q_{12b} & q_{20b} \end{bmatrix}^T \tag{27c}$$

c) The reciprocal polynomial model: Based on Equation 26, a version fulfilling the reciprocity condition was developed and is given in Equation 28.

$$\psi_{dm,r} = d_{10r} i_d + d_{11r} i_d i_q + d_{02r} i_q^2 + \frac{1}{3} q_{12r} i_q^3 \tag{28a}$$

$$\psi_q = q_{01r} i_q + q_{02r} i_q^2 + q_{12r} i_d i_q^2 + \frac{1}{2} d_{11r} i_d^2 + 2 d_{02r} i_d i_q \tag{28b}$$

The reciprocal formula can be linearized into Equation 29.

$$\boldsymbol{Y_r} = \begin{bmatrix} \boldsymbol{\psi_{dm,r}} \\ \boldsymbol{\psi_{q,r}} \end{bmatrix} \tag{29a}$$

$$\boldsymbol{X_r} = \begin{bmatrix} \boldsymbol{i_d} & \boldsymbol{i_d i_q} & \boldsymbol{i_q^2} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\frac{1}{3} i_q^3} \\ \boldsymbol{0} & \boldsymbol{\frac{1}{2} i_d^2} & \boldsymbol{2 i_d i_q} & \boldsymbol{i_q} & \boldsymbol{i_q^2} & \boldsymbol{i_d i_q^2} \end{bmatrix} \tag{29b}$$

$$\boldsymbol{B_r} = \begin{bmatrix} d_{10r} & d_{11r} & d_{02r} & q_{01r} & q_{02r} & q_{12r} \end{bmatrix}^T \tag{29c}$$

4) The exponential model: By the formulas proposed by [4] in Equation 11, the unknown coefficients a_{d0} a_{d0}, a_{dd}, a_{q0}, a_{qq}, a_{dq} and unknown exponents α, β, γ, δ, had to be found by applying linear least square method to the measured values. The equations were redefined with the defined variable ψ_{dm} into Equation 30. By this measure, the calculated currents are independent of the permanent magnet flux linkage. Or in other words, the permanent magnet flux linkage is defined as independent of the applied d- and q-axis currents.

$$i_{d,e} = (a_{d0} + a_{dd} |\psi_{dm}|^\alpha + \frac{a_{dq}}{\delta + 2} |\psi_{dm}|^\gamma |\psi_q|^{\delta+2}) \psi_{dm} \tag{30a}$$

$$i_{q,e} = (a_{q0} + a_{qq} |\psi_q|^\beta + \frac{a_{dq}}{\gamma + 2} |\psi_{dm}|^{\gamma+2} |\psi_q|^\delta) \psi_q \tag{30b}$$

The linearization of Equation 30 can be seen in Equation 31-35.

$$\boldsymbol{x} = |\boldsymbol{\psi_{dm}}|^\alpha \boldsymbol{\psi_{dm}} \tag{31}$$

$$\boldsymbol{y} = \frac{1}{\delta + 2} |\boldsymbol{\psi_{dm}}|^\gamma |\boldsymbol{\psi_q}|^{\delta+2} \boldsymbol{\psi_{dm}} \tag{32}$$

$$\boldsymbol{z} = |\boldsymbol{\psi_q}|^\beta \boldsymbol{\psi_q} \tag{33}$$

$$\boldsymbol{w} = \frac{1}{\gamma + 2} |\boldsymbol{\psi_{dm}}|^{\gamma+2} |\boldsymbol{\psi_q}|^\delta \boldsymbol{\psi_q} \tag{34}$$

$$\boldsymbol{Y_e} = \begin{bmatrix} \boldsymbol{i_{d,e}} \\ \boldsymbol{i_{q,e}} \end{bmatrix} \tag{35a}$$

$$\boldsymbol{X_e} = \begin{bmatrix} \boldsymbol{\psi_{dm}} & \boldsymbol{0} & \boldsymbol{x} & \boldsymbol{0} & \boldsymbol{y} \\ \boldsymbol{0} & \boldsymbol{\psi_q} & \boldsymbol{0} & \boldsymbol{z} & \boldsymbol{w} \end{bmatrix} \tag{35b}$$

$$\boldsymbol{B_e} = \begin{bmatrix} a_{d0} & a_{q0} & a_{dd} & a_{qq} & a_{dq} \end{bmatrix}^T \tag{35c}$$

The vectors x, y, z, w are not calculated as vector products but by multiplying the variables element-wise, meaning these vectors have the same size as the flux linkage vectors.

B_e was solved by Equation 23, incorporated in the MATLAB function *regress*. Initially all exponents, α, β, γ, δ, were assumed to equal zero. The same linear least square system was then iteratively recalculated but with different combinations of integer values for the exponents, from 0 to 9. Finally, the combination of values for the exponents giving the smallest residuals to the curve fit was selected for the exponential model.

D. Plotting the proposed formulas

1) Plotting the FE calculated data: Having vectors of measured d- and q-axis flux linkages and currents, the vectors had to be restructured such that the flux linkages could be plotted in a 3D plot for a defined 2D grid formed by the d- and q-axis currents. To achieve this, the MATLAB function *meshgrid* was used to create a 2D grid with 40 equally spaced coordinates ranging from 0 to $-2pu$ for the d-axis current and 0 to $2pu$ for the q-axis current. By *griddata*, the scattered data of either $\boldsymbol{\psi_d}$ or $\boldsymbol{\psi_q}$ in relation to the scattered $\boldsymbol{i_d}$ and $\boldsymbol{i_q}$ data points were structured by linear interpolation to a grid corresponding to the defined 2D current grids. In short, the one-dimensional d- and q-axis currents and flux linkages vectors were redefined to 3D grids, such that values of either ψ_d or ψ_q could be plotted along the z-axis while having i_d along the x-axis and i_q along the y-axis. The complete MATLAB script to achieve this can be found in Appendix C-A. It should be noted that these grids can directly be used as look-up tables in the control system, as selected combinations from the i_d- and i_q-grids would result in unique values of either ψ_d or ψ_q, or opposite if i_d and i_q are plotted for grids of ψ_d and ψ_q. The calculated grids of values were plotted in a surface plot using the function *surf*, as seen in the Appendix C-B.

Having developed a method to plot the FE calculated flux linkages in relation to the applied currents, the solved flux linkages by the different curve fitting magnetic models could be evaluated by plotting them together with the FE calculated flux linkages.

2) Plotting curve fitted models: For the linear magnetic model formulas and the polynomial formulas, the flux linkage grids were calculated by using the model formulas, the coefficients found by curve fitting and the defined 2D current grids. The flux linkages were then plotted as surface plots together with the surface plot of the FE calculated ones, as seen in Appendix C-E to C-H.

The exponential model was defined such that the d- and q-axis currents were calculated for given d- and q-axis flux linkages. Thus, in order to plot the flux linkage surfaces, a new set of $\boldsymbol{i_d}$ and $\boldsymbol{i_q}$ vectors were calculated using the curve fitted coefficients and exponents and the FE calculated ψ_{dm} and ψ_q vectors from COMSOL. Having calculated the new current vectors as a function of the FE calculated flux linkage vectors, the currents and flux linkages were plotted in the same manner as for the initial measured data, as explained in Section III-D1. The complete curve fitting and plotting of the exponential model can be found in Appendix C-I.

Before plotting the d-axis flux linkage for the polynomial and exponential models, the ψ_m term was added to the calculated ψ_{dm}, such that the surface plots were plotted for ψ_d.

E. Model performance evaluation

When comparing the performance of the different proposed magnetic models, three main evaluations were done: 1) The numerical difference between the model estimated and FE calculated flux linkage surfaces within the operating regions, 2) the models' ability to extrapolate flux linkage data outside of the area they were curve fitted to and 3) the models' ability to be curve fitted with very few input data points.

1) Difference between the FE calculations and the model estimations: In order to compare the performance of the different magnetic models, the model developed flux linkage surfaces were subtracted from the FE calculated flux linkages. The different models were then compared in terms of how small the difference between the model estimated and FE calculated flux linkages were within the typical operating region of the motor, as explained in Figure 6.

2) Data extrapolation outside of the curve fitting area: For this analysis, firstly, the FE calculated current and flux linkage values outside of the region $i_d \in [-1, 0]$ and $i_q \in [0, 1]$ were deleted, such that the coefficients of the formulas were fitted to this limited region. Secondly, the flux linkages were plotted in the range of $i_d \in [-2, 0]$ and $i_q \in [0, 2]$, to see how well the curve fitted flux linkages fitted the COMSOL solved flux linkages outside of the region they had been curve fitted to.

3) Curve fitting by few input data points: It is beneficial if the relation between the d- and q-axis flux linkages and currents can be accurately modeled by curve fitting to few data points, especially if the FE model of the machine is not available. As it is time-consuming to measure the flux linkages and currents for a physical machine, it would be a great advantage if the model could be curve fitted for a few operating points. This would be especially valuable for the plug-and-play methods, where magnetic models are developed during start up of an unknown motor. To evaluate the models' ability to curve fit for few operating points accurately, only 9 FE calculated data points were used to calculate the model coefficients and exponents by the linear least square method. The 9 operating points were combinations of i_d from $0pu$ to $-2pu$

with steps of $-1pu$ and i_q from $0pu$ to $2pu$ with steps of $1pu$.

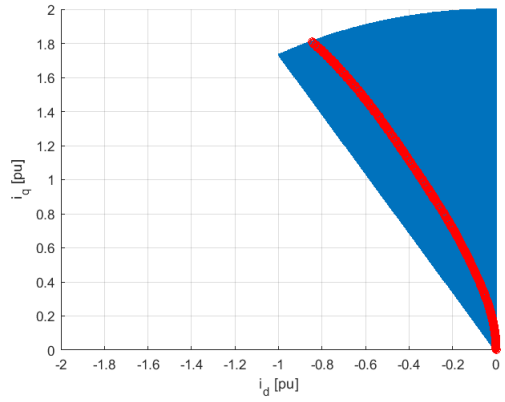
IV. Method: Control improvement

In order to demonstrate the use of the improved magnetic models when controlling the motor, the current references providing MTPA control, meaning the MTPA trajectory, were derived using both the linear model and the best polynomial model. These current references were compared to the combination of d- and q-axis currents the FE model calculated to give maximum torque per ampere for each total stator current. This COMSOL calculated MTPA trajectory was considered to be the ideal MTPA trajectory, such that the MTPA trajectories derived from the linear model and the best polynomial model were evaluated in terms of how close they were to the FE calculated MTPA trajectory.

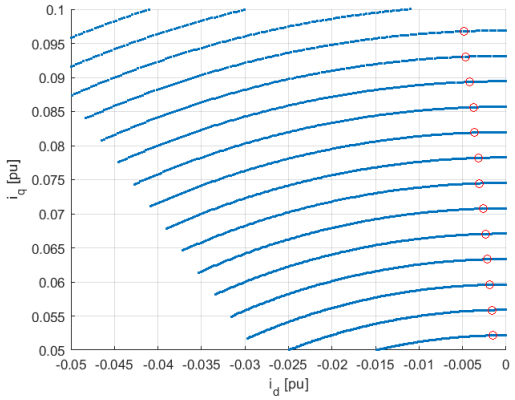
In addition, based on the analysis of the MTPA trajectories, deviations in the analytically calculated torques were observed when calculating the torque based on the magnetic models. Thus, the increased accuracy of the analytically calculated torque was evaluated when using the best polynomial model compared to the linear model. Finally, as there still was a difference between the analytically calculated torque by the best polynomial model and the FE calculated torque by Arkkio's method, the analytical torque equation itself was evaluated based on the FE calculated flux linkages and currents in COMSOL.

A. FE calculated MTPA trajectory

Using the same IPMSM model as in Section III, the torque and d- and q-axis flux linkages could be calculated for different d- and q-axis currents. Instead of predefining combinations of I_d and I_q currents, the currents were defined by the load angle θ and the total stator current amplitude I_s. The COMSOL model was solved for different loading angles and total stator currents, such that the MTPA trajectory could be derived by finding the θ providing maximum torque for each total stator current. The torque was calculated using Arkkio's method, and the measurements were done with sweeps of I_s from $1e-6A$ to $2\cdot\sqrt{2}\cdot 2.85A$ with steps of $0.015A$. The load angle, θ, was only swept from $\pi/2rad$ to $2\pi/3rad$ with steps of $\pi/1000$, as the optimal load angle to achieve maximum torque was known to be in this region by experience. A MATLAB script was built to extract the load angle, which gave the highest torque for each selected total stator current. The MATLAB script is given in Appendix C-J and a plot of the MATLAB selected d- and q-axis current references corresponding to the calculated load angles providing maximum torque for each simulated total stator current is given in Figure 12.



(a)



(b)

Fig. 12: (a) The MTPA trajectory found numerically by calculating the torque in COMSOL for different stator currents and load angles, such that all the i_d, i_q combinations ensuring maximum torque for any total stator current, i_s, can be found. (b) Zoom of (a).

B. Analytical MTPA trajectories

1) Linear model: The analytical torque equation when assuming linear magnetics is given in Equation 12. As explained in II-C, the torque equation can be rewritten as a function of i_q and i_s by using the relation from Equation 3b, as given in Equation 36.

$$\tau_e = \psi_m i_q + (x_q - x_d)\sqrt{i_s^2 - i_q^2}\, i_q \tag{36}$$

In order to find the extremal point of the function, meaning for which i_q the maximum torque is obtained, the derivative of Equation 36 with respect to i_q was solved, as shown in Equation 37.

$$\frac{\partial \tau_e}{\partial i_q} = \psi_m + (x_q - x_d)\frac{i_s^2 - 2i_q^2}{\sqrt{i_s^2 - i_q^2}} \tag{37}$$

Setting Equation 37 equal to zero and reinserting the expression for i_d from Equation 2 results to Equation 38.

$$i_q = \pm\sqrt{i_d^2 - \frac{\psi_m i_d}{x_q - x_d}} \tag{38}$$

Reinserting Equation 38 into Equation 3a, an explicit function for the i_d providing the maximum torque for any total stator current, i_s, could be expressed as in Equation 39.

$$i_d = \frac{\psi_m}{4(x_q - x_d)} - \sqrt{(\frac{\psi_m}{4(x_q - x_d)})^2 + \frac{i_s^2}{2}} \tag{39}$$

Having calculated the d-axis current, the corresponding q-axis current could be calculated by Equation 3b for the chosen total stator current. The optimal i_d and i_q combinations giving maximum torque for any selected total stator current could then be plotted.

2) Best polynomial model: The analytical torque equation given in Equation 7, was expanded into Equation 40 by the formulas of the best polynomial model, as given in Equation 26.

$$\begin{aligned}\tau_e = {} & (d_{10b} i_d + d_{11b} i_d i_q + d_{02b} i_q^2 + \psi_m) i_q \\ & - (q_{01b} i_q + q_{02b} i_q^2 + q_{12b} i_d i_q^2 + q_{20b} i_d^2) i_d\end{aligned} \tag{40}$$

By using the relation from Equation 3b, the torque equation could be written as a function with only i_d as unknown parameter, as given in Equation 41.

$$\begin{aligned}\tau_e = {} & (d_{10b} i_d + d_{11b} i_d \sqrt{i_s^2 - i_d^2} + d_{02b}(i_s^2 - i_d^2) \\ & + \psi_m)\sqrt{i_s^2 - i_d^2} - (q_{01b}\sqrt{i_s^2 - i_d^2} + q_{02b}(i_s^2 - i_d^2) \\ & + q_{12b} i_d (i_s^2 - i_d^2) + q_{20b} i_d^2) i_d\end{aligned} \tag{41}$$

Furthermore, the derivative of the torque with respect to i_d was solved into Equation 42.

$$\begin{aligned}\frac{\partial \tau_e}{\partial i_d} = {} & -\frac{2 d_{10b} i_d^2}{\sqrt{i_s^2 - i_d^2}} - \frac{\psi_m i_d}{\sqrt{i_s^2 - i_d^2}} - 3 d_{02b} i_d \sqrt{i_s^2 - i_d^2} \\ & + \frac{d_{10b} i_s^2}{\sqrt{i_s^2 - i_d^2}} - 3 d_{11b} i_d^2 + d_{11b} i_s^2 + 4 q_{12b} i_d^3 \\ & + \frac{2 q_{01b} i_d^2}{\sqrt{i_s^2 - i_d^2}} - \frac{q_{01b} i_s^2}{\sqrt{i_s^2 - i_d^2}} + 3 q_{02b} i_d^2 - 3 q_{20b} i_d^2 \\ & - 2 q_{12b} i_d i_s^2 - q_{02b} i_s^2\end{aligned} \tag{42}$$

In order to find the extremal point of the torque equation as a function of i_d, the nonlinear equation $\partial \tau_e / \partial i_d = 0$, had to be solved numerically. The *Trust Region Dogleg Method*, which is an extension of the *Gauss-Newton Method*, was used by the MATLAB function *fsolve*, to solve the equation. Having solved for the i_d giving the maximum torque for any i_s, the corresponding i_q was found by Equation 3b. It should be noted that the derivative of the torque with respect to the d-axis current can be written in different ways, and not all of them converged by the *fsolve* function.

C. Analytical MTPA trajectory comparison using COMSOL

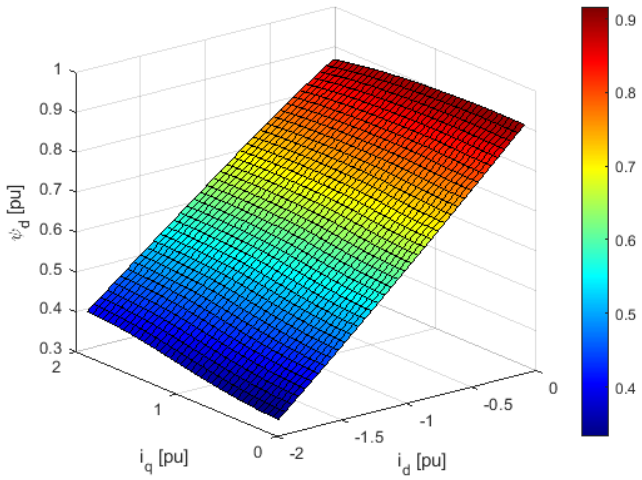
Even though the analytical MTPA trajectories were evaluated in terms of how well they corresponded to the numerically calculated one by COMSOL, it was also of interest to assess the difference in torque per ampere the two analytical MTPA trajectories actually would give. Thus, both of the proposed MTPA trajectories, meaning the i_d and i_q current references for different total stator currents, were exported and run through the COMSOL model. The FE calculated torques per ampere using the current references developed from the linear model and the best polynomial model were exported back to MATLAB and plotted, as given by the MATLAB script in Appendix C-K.

D. Analytical torque calculation

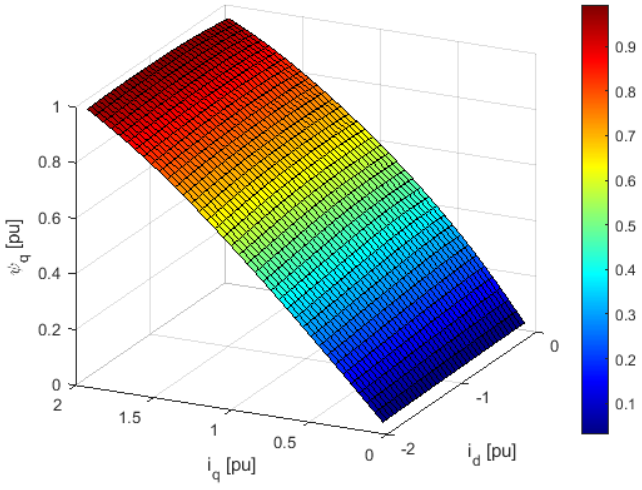
If there is no torque meter connected to the motor, the motor control system would estimate the torque of the motor based on the measured currents and calculated flux linkages from the magnetic model. The estimated torque would then influence whether the control system would judge if the torque reference had been reached or not. Thus, for selected current references and corresponding FE calculated torque, it is of interest to evaluate if the best polynomial model would be able to estimate better the FE calculated torque than the linear model.

E. Torque equation evaluation

As will be shown, the MTPA trajectory proposed when using the best polynomial model did not entirely correspond to the MTPA trajectory by COMSOL. In addition, when calculating the torque analytically using the best polynomial model by Equation 40, the calculated torque did not equal the FE calculated torque. Thus, both the d- and q-axis flux linkages and currents were measured in COMSOL using the current references derived for the best polynomial model. The torque was then calculated using the analytical torque equation from Equation 7 as well as the FE calculated torque by Arkkio's method. The accuracy of the torque equation was then evaluated based on the plot of the two derived torques, as given by the MATLAB script in Appendix C-K.

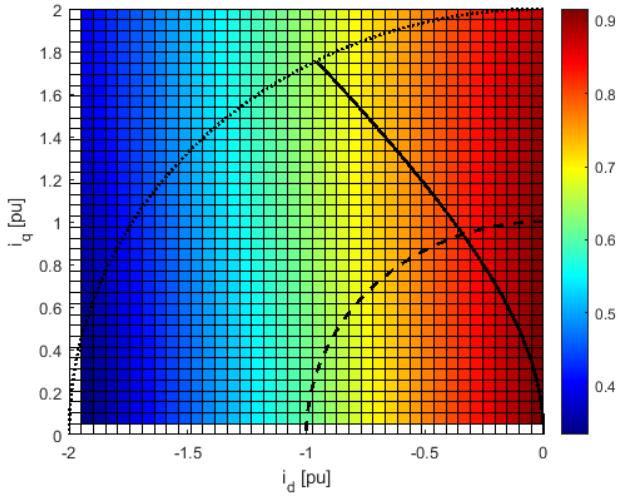


(a)

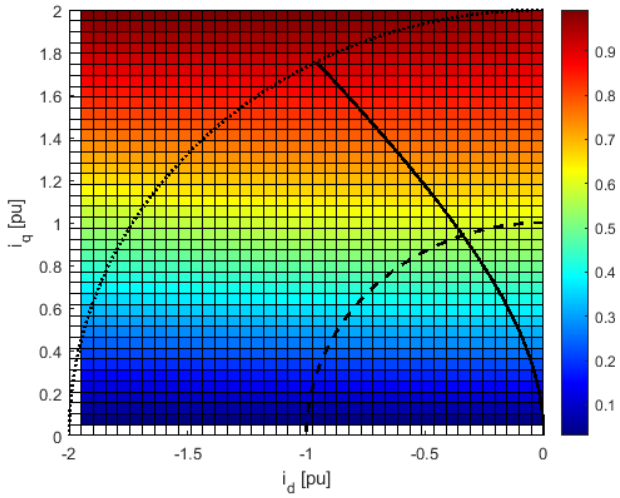


(b)

Fig. 13: 3D plot of the measured ψ_d and ψ_q by COMSOL for combinations of i_d and i_q currents.



(a)



(b)

Fig. 14: 2D plot of the measured ψ_d and ψ_q by COMSOL for combinations of i_d and i_q currents. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked in the plot to emphasize the operating region of the motor.

V. Results: Magnetic modeling

A. Flux linkage curves from COMSOL

A surface plot interpolating between the FE calculated data points of the d- and q-axis flux linkages in relation to the applied currents using COMSOL can be found in Figure 13 and Figure 14. The figures show that the d-axis flux linkage is not zero when i_d, i_q are zero, due to the permanent magnet. However, when negative d-axis current is applied, the d-axis flux linkage gets reduced. In addition, there can be observed increased d-axis flux linkage for positive i_q currents, meaning there are cross-coupling phenomena present.

Figure 13 and Figure 14 also show how the q-axis flux linkage is zero for no stator currents, as the permanent magnets produce no flux in the q-axis direction. It can be observed that the surface bends for high negative i_d currents, meaning there is cross-coupling.

B. Comparing magnetic models to the FE calculated values

All the coefficients and exponents calculated for all the proposed formulas by the linear least square curve fitting method can be found in Table IV. In the following 3D plots, the color of the COMSOL calculated surface plots of the flux linkages will be set to red when plotting them together with the surface plots of the curve fitted magnetic models to better be able to differentiate the plots.

The percentage deviation between the measured and curve fitted surfaces at the operating points along the MTPA locus for stator currents of $1pu$ and $2pu$ have been calculated and can be found in Table V. Stator current of $1pu$ corresponds to full loading of the motor, while $2pu$ corresponds to a transient loading state with considerably saturation and cross-coupling.

1) Linear model: As x_d was calculated when applying 10% positive i_d current, the flux linkages for small i_d values are quite correctly estimated. However, the deviation increases when applying i_d currents lower than $-1pu$. For the FE calculated curve of the q-axis flux linkage, the slope is decreasing for higher i_q currents due to saturation. Thus, the deviation between the FE computed and model estimated flux linkages by the linear model are increasing for higher i_q currents.

It should be noted that for higher negative i_d currents, ψ_d modeled by constant x_d is underestimating the the actual measured ψ_d. For ψ_q the opposite effect is present, meaning for higher positive i_q currents the modeled ψ_q by constant x_q is overestimating the actual q-axis flux linkage in the machine, as seen in Figure 15 and Figure 16.

2) Simple polynomial model: Compared to the d-axis flux linkage when using the linear model, the difference between the FE computed and curve fitted surfaces by the simple polynomial model is smaller within the whole operating region of the machine, and especially along the MTPA trajectory, as seen in Figure 17 and Figure 18. However, within the operating regions, when having stator currents less than $1pu$, the linear model is more accurate than the simple polynomial formulas for the d-axis flux linkage.

For the q-axis flux linkage, by introducing a second-order term to the simple polynomial model, it can be observed in Figure 17 and Figure 18 that the difference between the FE calculated and curve fitted q-axis flux linkage is drastically reduced, compared to the linear model.

3) Best polynomial model: After having found the unknown coefficients of the best polynomial model using curve fitting, the corresponding plotted surfaces prove small differences to the FE calculated ψ_d and ψ_q surfaces, as seen in Figure 19 and Figure 20. In Table V, it can be seen how the best polynomial model provides a relative difference of less than 1% to the FE computed flux linkages for critical operating points at MTPA control and 1pu and 2pu applied stator currents. However, the coefficients to be found using the linear least square method have increased from 3 to 7, compared to the simple polynomial model.

4) Reciprocal polynomial model: By forcing the best polynomial model to fulfill the reciprocity condition, the number of unknown coefficients to be estimated was reduced from 7 to 6. As seen in Figure 21 and Figure 22, the curve fitted reciprocal model performed better than the best polynomial model for the operating regions within $1pu$ stator current. When summing up the residuals of the curve fit for the reciprocal and best polynomial models, the residuals of the best polynomial model were smaller than the ones of the reciprocal polynomial model. In addition, the numerical values of the coefficients of the reciprocal formulas were more dependent on which region the formulas were curve fitted to than the best polynomial model due to the modeled interdependence of the flux linkages for the reciprocal polynomial model. Thus, for the MTPA analysis, the best polynomial model was chosen instead of the reciprocal polynomial model.

5) Exponential model: When curve fitting the exponential model to the FE computed flux linkages, the curve fit was done for all combinations of the exponents from 0 to 9 to find the combination of exponents giving the smallest residuals. The optimal curve fit was found for an a_{dd} parameter with a negative value, but as one of the requirements for the exponential formulas was only to have positive exponents and parameters, a_{dd} was forced to be zero.

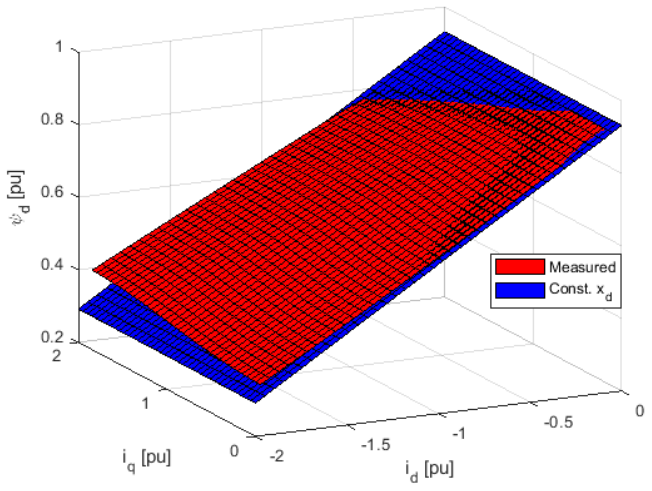
The 3D surface plots can be found in Figure 23, while the

TABLE IV: Coefficients and exponents for the proposed models after being curve fitted to the FE calculated data. The curve fitting was done for 26244 measurement points in $i_d \in [-2, 0]$ and $i_q \in [0, 2]$.

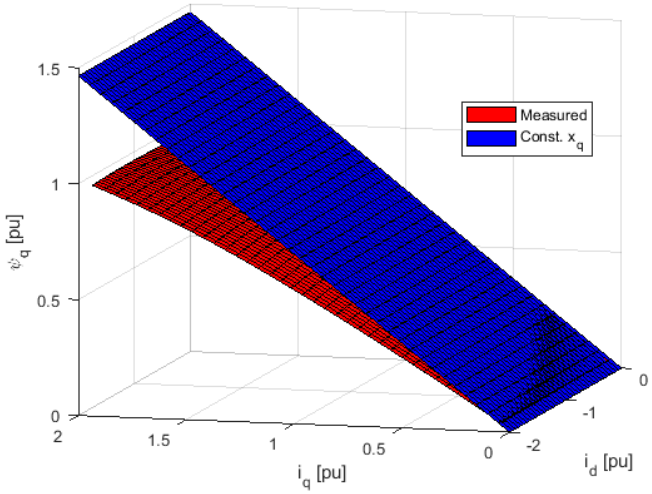
Parameter	Value [pu]
Linear	
x_d	0.3200
x_q	0.7324
Simple polynomial	
d_{10s}	0.2923
q_{01s}	0.6917
q_{02s}	-0.0949
Best polynomial	
d_{10b}	0.3168
d_{11b}	-0.0321
d_{02b}	-0.0076
q_{01b}	0.7196
q_{02b}	-0.1130
q_{12b}	-0.0065
q_{20b}	-0.0105
Reciprocal polynomial	
d_{10r}	0.3141
d_{11r}	-0.0254
d_{02r}	0.0024
q_{01r}	0.7196
q_{02r}	-0.1130
q_{12r}	-0.0065
Exponential	
α	2
β	2
γ	0
δ	1
a_{d0}	3.2603
a_{dd}	0
a_{q0}	1.4666
a_{qq}	0.4845
a_{dq}	1.9733

TABLE V: Deviations from the model estimated flux linkages to the FE calculated ones, relative to the FE calculated ones, for operating points of 1pu and 2pu stator current while operating with MTPA control. The magnetic modeling was done for 26244 measurement points in $i_d \in [-2, 0]$ and $i_q \in [0, 2]$.

MTPA, $i_s = 1pu$		MTPA, $i_s = 2pu$	
$\lvert\Delta\psi_d\rvert$ [%]	$\lvert\Delta\psi_q\rvert$ [%]	$\lvert\Delta\psi_d\rvert$ [%]	$\lvert\Delta\psi_q\rvert$ [%]
Constant x_d, x_q			
0.8771	17.8547	5.6053	38.3557
Simple polynomial			
0.3942	3.6999	1.2757	1.4094
Best polynomial			
0.2302	0.8826	0.3349	0.2662
Reciprocal polynomial			
0.2731	0.3451	0.2326	0.2071
Exponential			
0.1584	0.9185	0.5956	0.1729

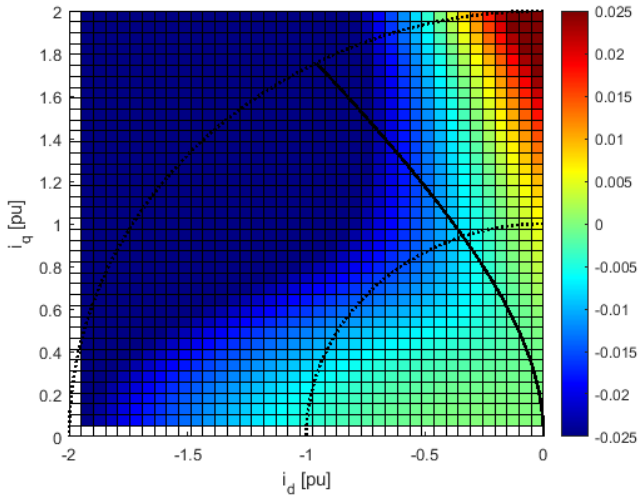


(a)

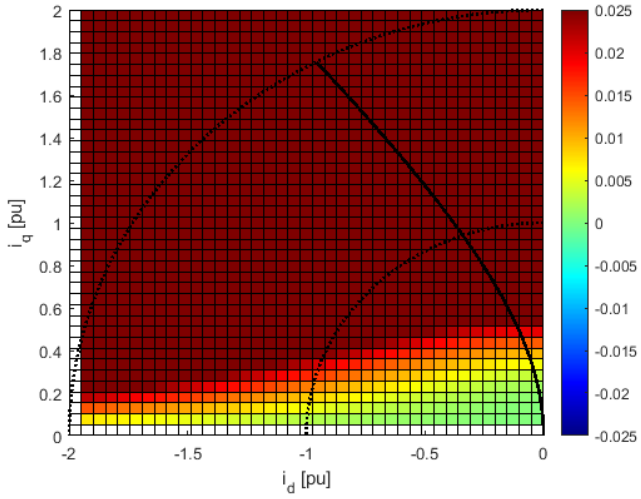


(b)

Fig. 15: The FE calculated (a) ψ_d and (b) ψ_q by COMSOL, plotted with the calculated ψ_d and ψ_q when using the linear model, meaning assuming constant x_d, x_q.

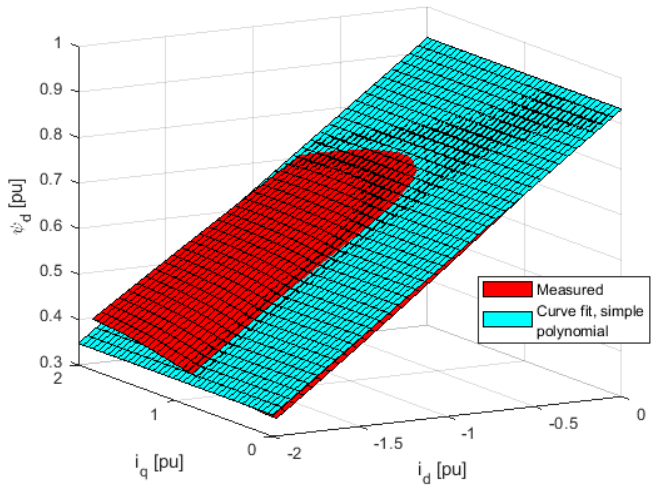


(a)

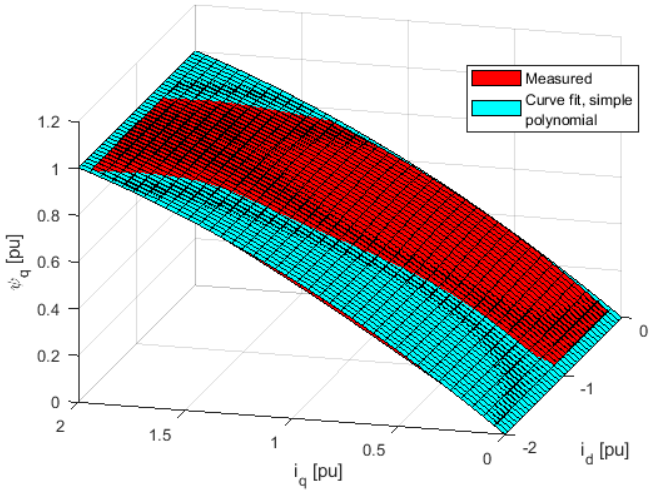


(b)

Fig. 16: The difference between the FE calculated and the model estimated flux linkages by the linear model, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

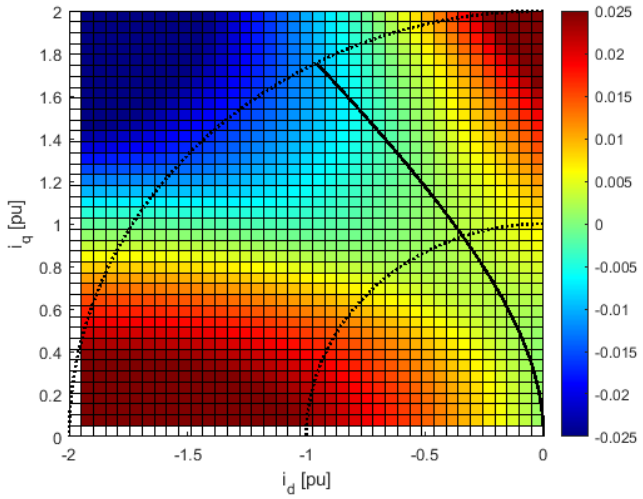


(a)

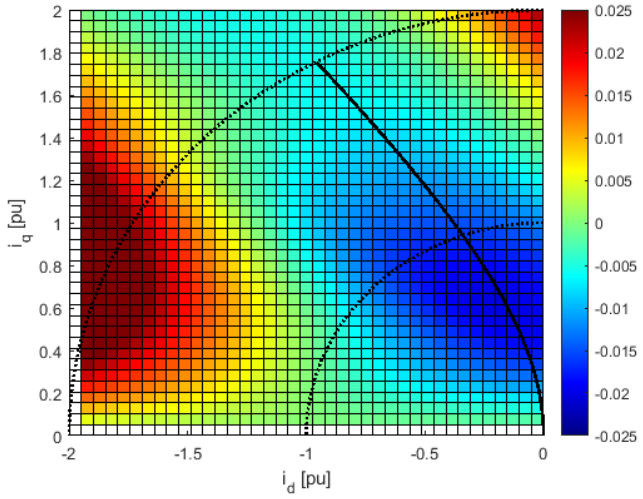


(b)

Fig. 17: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the simple polynomial model with coefficients curve fitted for 26244 measurement points in $i_d \in [-2, 0]$ and $i_q \in [0, 2]$.

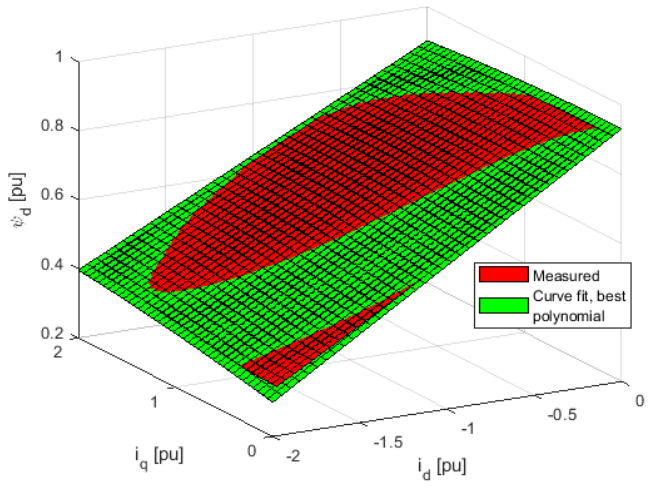


(a)

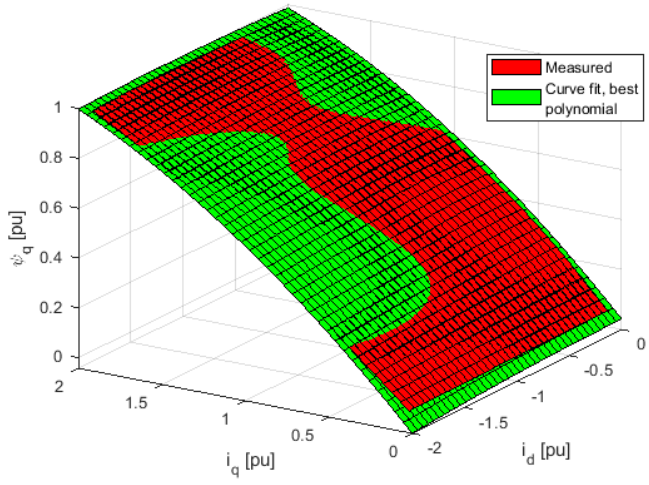


(b)

Fig. 18: The difference between the FE calculated and the model estimated flux linkages by the simple polynomial model, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

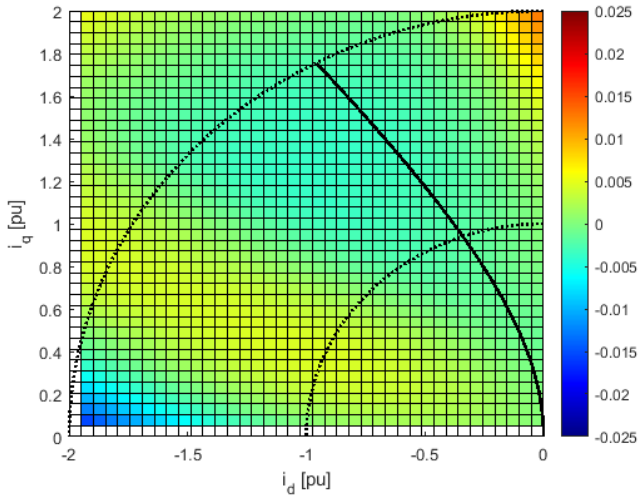


(a)

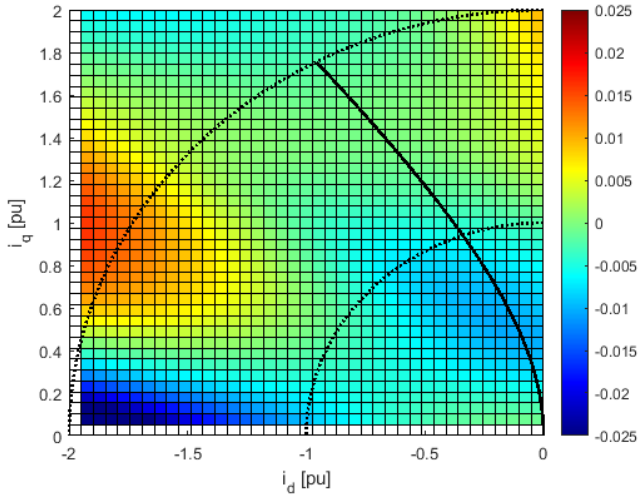


(b)

Fig. 19: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the best polynomial model with coefficients curve fitted for 26244 measurement points in $i_d \in [-2, 0]$ and $i_q \in [0, 2]$.

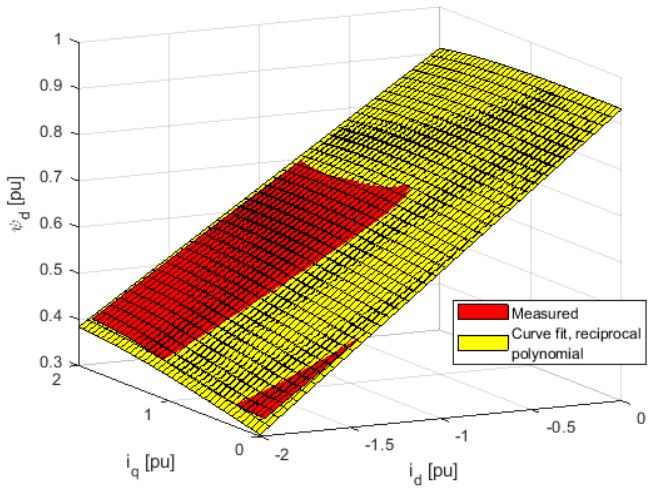


(a)

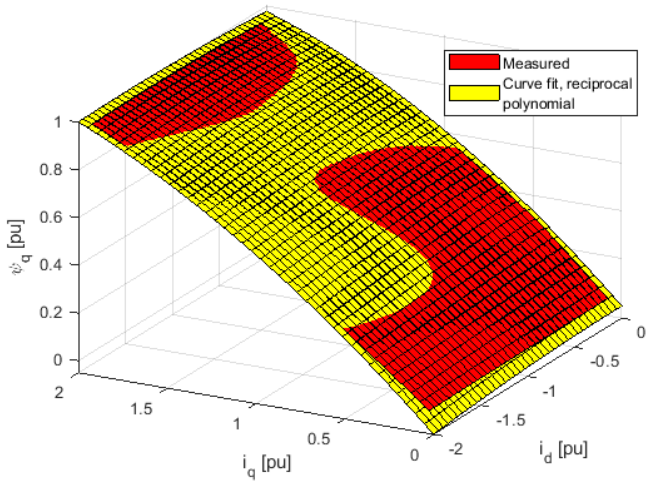


(b)

Fig. 20: The difference between the FE calculated and the model estimated flux linkages by the best polynomial model, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

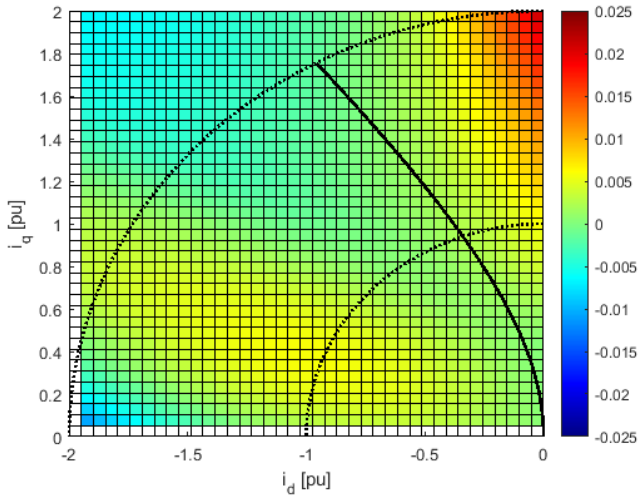


(a)

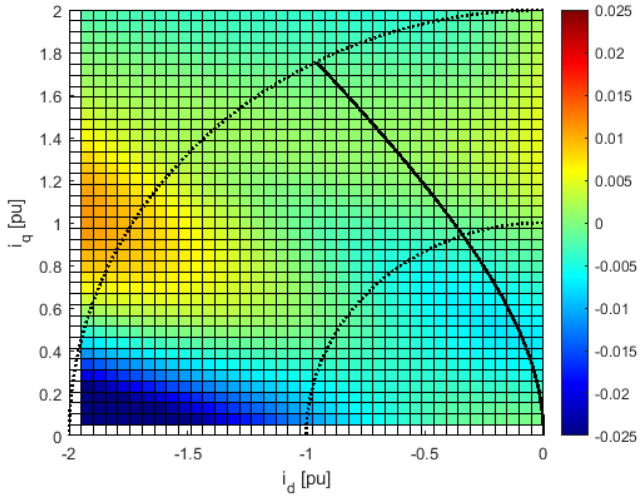


(b)

Fig. 21: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the reciprocal polynomial model with coefficients curve fitted for 26244 measurement points in $i_d \in [-2, 0]$ and $i_q \in [0, 2]$.

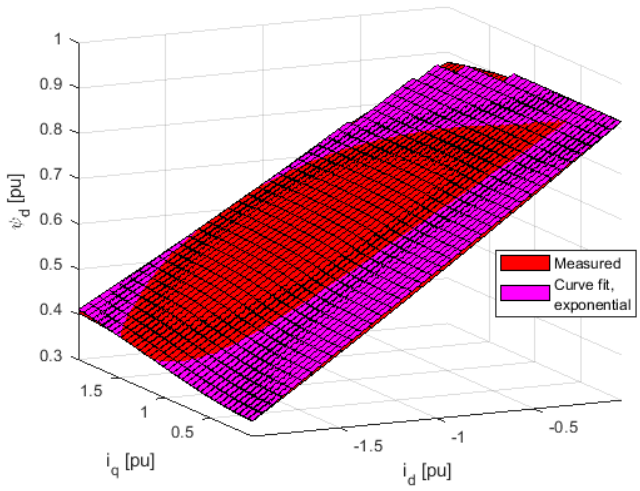


(a)

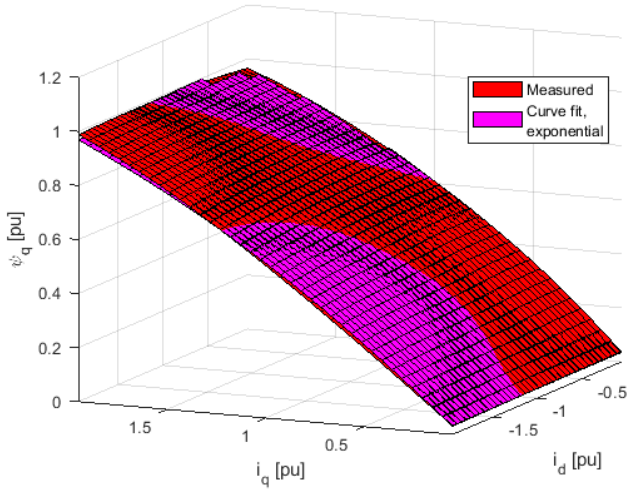


(b)

Fig. 22: The difference between the FE calculated and the model estimated flux linkages by the reciprocal polynomial model, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

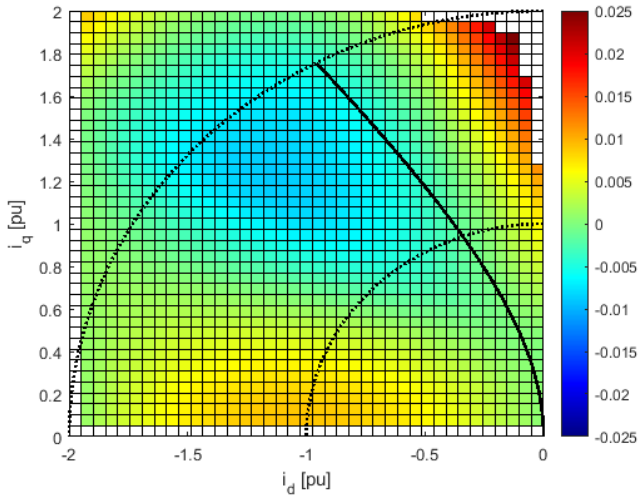


(a)

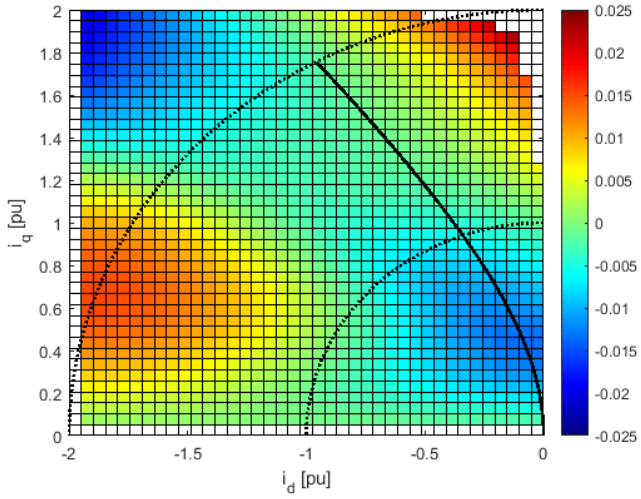


(b)

Fig. 23: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the exponential model with coefficients and exponents curve fitted for 26244 measurement points in $i_d \in [-2, 0]$ and $i_q \in [0, 2]$.



(a)



(b)

Fig. 24: The difference between the FE calculated and the model estimated flux linkages by the exponential model, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

2D surface plot can be found in Figure 24. As mentioned in Section II-B2, the exponential model calculates the d- and q-axis currents based on ψ_d and ψ_q. Thus when plotting the calculated currents to the defined i_d, i_q grids, the calculated currents of the exponential formulas did not fit within the defined grid, leading to areas without data for the 3D and 2D surface plots. However, the plotted areas without measurement data were outside of the operating region, meaning these data would have been neglected for the analysis either way. The exponential model also achieved a relative difference to the FE calculated flux linkages to less than 1% within the critical operating points of 1pu and 2pu total stator current and MTPA control, as given in Table V.

C. Data extrapolation outside of the curve fitting area

In this subsection, the calculated flux linkages were plotted up to $i_d \in [-2, 0]$ and $i_q \in [0, 2]$, while being curve fitted to the area of $i_d \in [-1, 0]$ and $i_q \in [0, 1]$. The relative deviation between the curve fitted models of the flux linkages and the FE calculated ones along the MTPA trajectory for stator currents of 1pu and 2pu can be found in Table VI. The surface plots of the best polynomial model can be found in Figure 25 and Figure 26, and the exponential model in Figure 27 and Figure 28. It can be seen that both models are more accurate within $i_d \in [-1, 0]$ and $i_q \in [0, 1]$ compared to when being curve fitted up to $i_d \in [-2, 0]$ and $i_q \in [0, 2]$. In addition, the best polynomial model is more accurate outside of the curve fitted area than the exponential model. The flux linkage plots of the simple and reciprocal models can be found in Appendix B.

D. Performance of curve fitting with only 9 measurement points

The previous curve fittings were done for 26244 measurement points. However, having so many measurement points is not feasible if the FE model is not available, and the flux linkages would have to be measured at the lab. Thus, in order to evaluate the usability of the magnetic models with input data from measurements at the lab, the curve fitting was done for only 9 measurement points. The surface plots for the best polynomial model can be found in Figure 29 and Figure 30 and for the exponential model in Figure 31 and Figure 32. The surface plots of the simple and reciprocal models can be found in the Appendix B. The deviations from the FE computed flux linkages for typical operating points can be found in Table VII.

VI. Results: Control improvement

A. MTPA trajectories

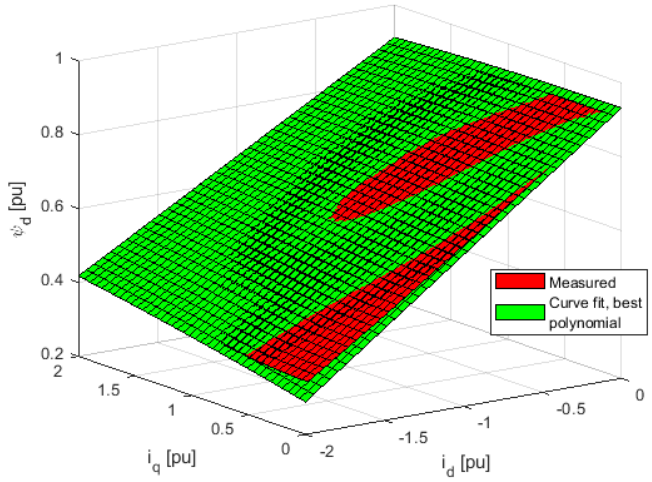
The empirically calculated MTPA trajectory from the COMSOL model was plotted together with the analytically calculated MTPA trajectories for the linear and best polynomial magnetic model, as seen in Figure 33. It can be observed that

TABLE VI: Deviations from the model estimated flux linkages to the FE calculated ones, relative to the FE calculated ones, for operating points of 1pu and 2pu stator current while operating with MTPA control. The magnetic modeling was done for 6561 measurement points in $i_d \in [-1, 0]$ and $i_q \in [0, 1]$.

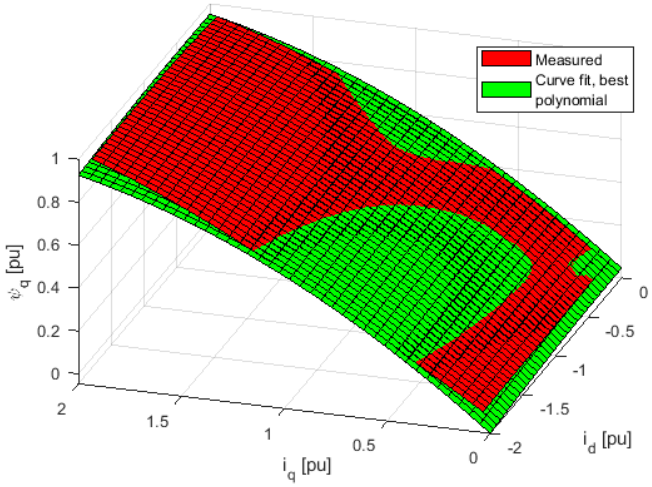
MTPA, $i_s = 1pu$		MTPA, $i_s = 2pu$	
$\lvert\Delta\psi_d\rvert$ [%]	$\lvert\Delta\psi_q\rvert$ [%]	$\lvert\Delta\psi_d\rvert$ [%]	$\lvert\Delta\psi_q\rvert$ [%]
Simple polynomial			
0.2658	0.2483	3.5233	1.8320
Best polynomial			
0.0211	0.0070	1.4515	3.9194
Reciprocal polynomial			
0.1167	0.0460	0.9553	2.9997
Exponential			
0.1704	0.2011	3.5808	2.3595

TABLE VII: Deviations from the model estimated flux linkages to the FE calculated ones, relative to the FE calculated ones, for operating points of 1pu and 2pu stator current while operating with MTPA control. The magnetic modeling was done for 9 measurement points in $i_d \in [-2, 0]$ and $i_q \in [0, 2]$.

MTPA, $i_s = 1pu$		MTPA, $i_s = 2pu$	
$\lvert\Delta\psi_d\rvert$ [%]	$\lvert\Delta\psi_q\rvert$ [%]	$\lvert\Delta\psi_d\rvert$ [%]	$\lvert\Delta\psi_q\rvert$ [%]
Simple polynomial			
0.3942	3.6999	1.2757	1.4094
Best polynomial			
0.2928	2.9967	1.1176	1.0250
Reciprocal polynomial			
0.4095	1.8703	0.4310	0.5981
Exponential			
0.1213	2.3208	1.1360	1.0489

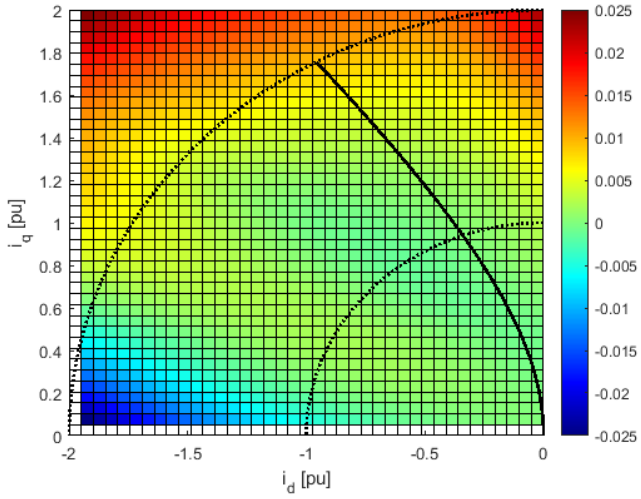


(a)

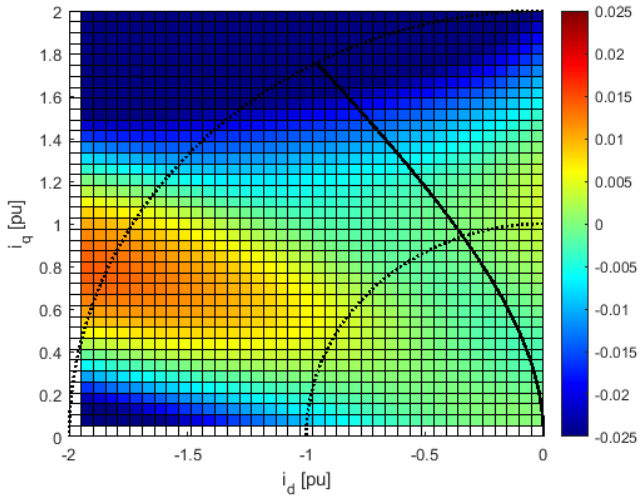


(b)

Fig. 25: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the best polynomial model with coefficients curve fitted for 6561 measurement points in $i_d \in [-1, 0]$ and $i_q \in [0, 1]$.

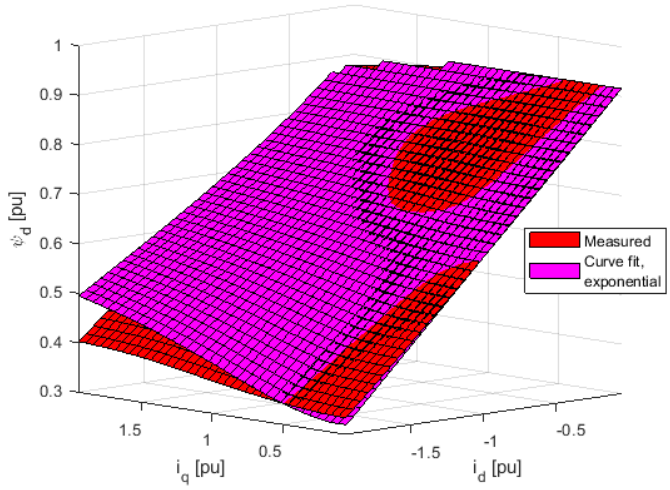


(a)

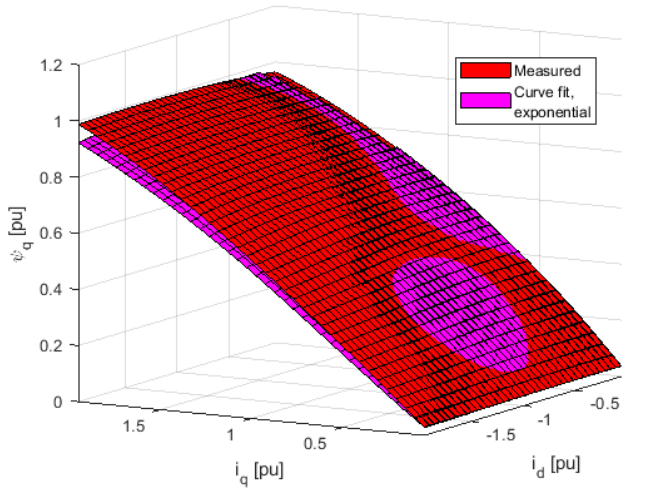


(b)

Fig. 26: The difference between the FE calculated and the model estimated flux linkages by the best polynomial model for 6561 measurement points between $i_d \in [-1, 0]$ and $i_q \in [0, 1]$, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

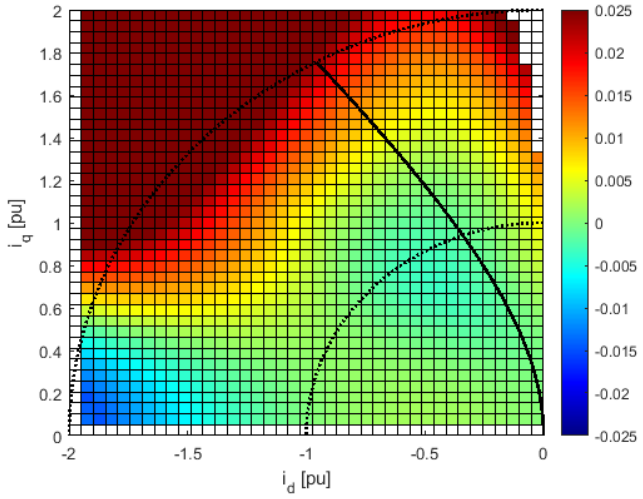


(a)

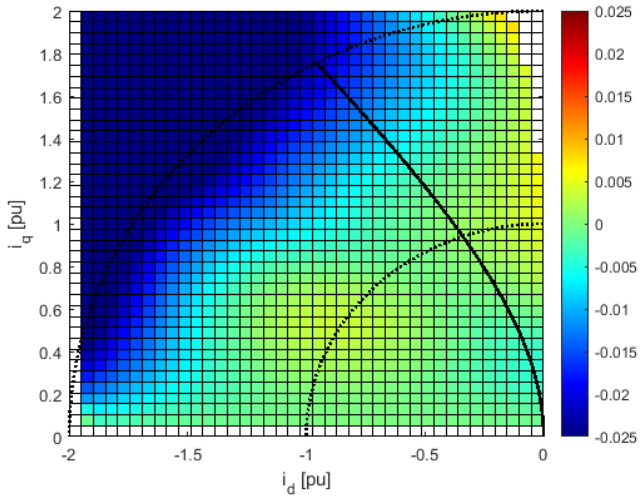


(b)

Fig. 27: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the exponential model with coefficients and exponents curve fitted for 6561 measurement points in $i_d \in [-1, 0]$ and $i_q \in [0, 1]$.

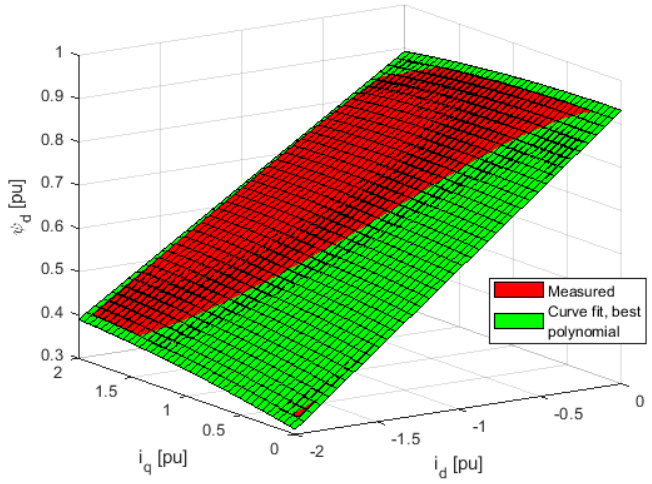


(a)

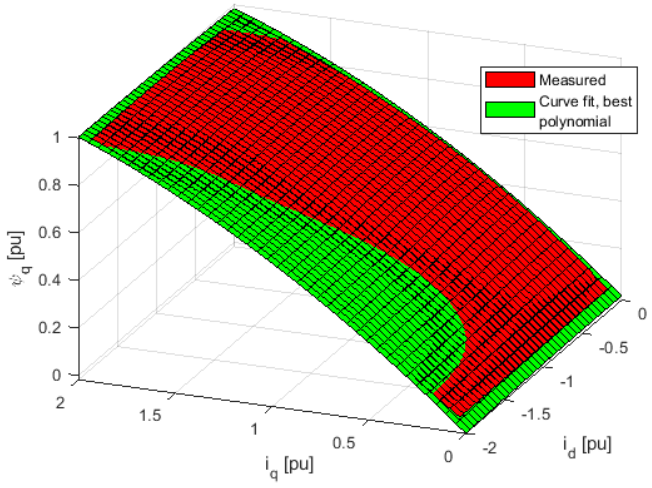


(b)

Fig. 28: The difference between the FE calculated and the model estimated flux linkages by the exponential model for 6561 measurement points between $i_d \in [-1, 0]$ and $i_q \in [0, 1]$, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

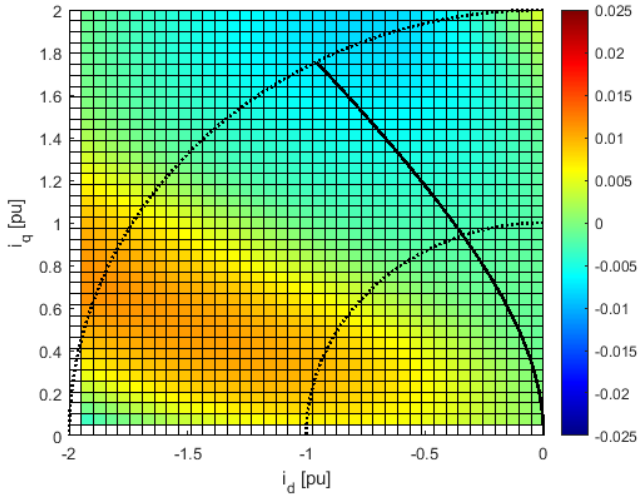


(a)

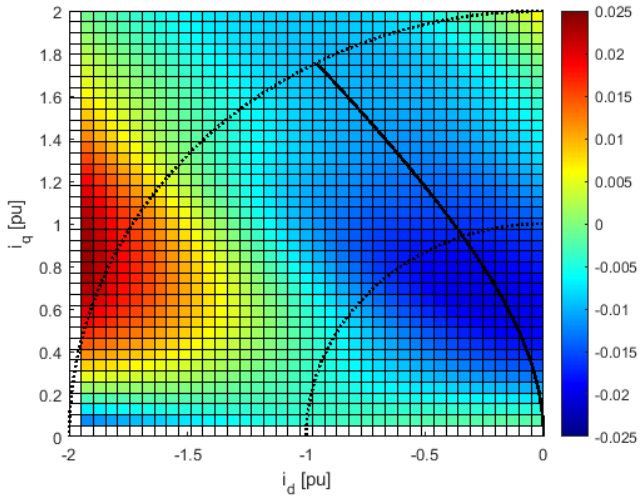


(b)

Fig. 29: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the best polynomial model with coefficients curve fitted for 9 measurement points in $i_d \in [-2, 0]$ and $i_q \in [0, 2]$.

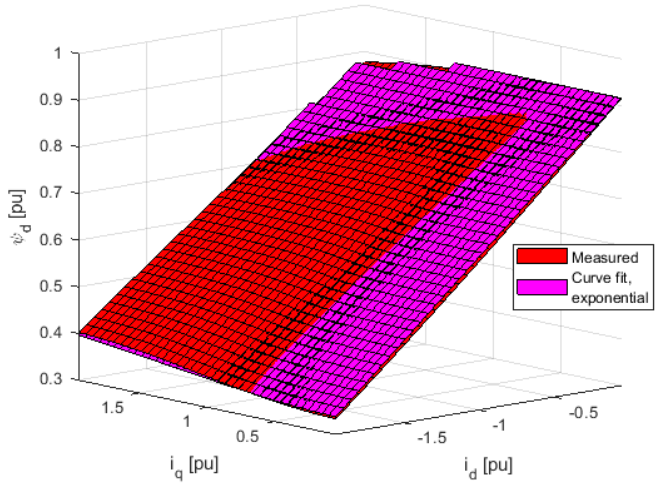


(a)

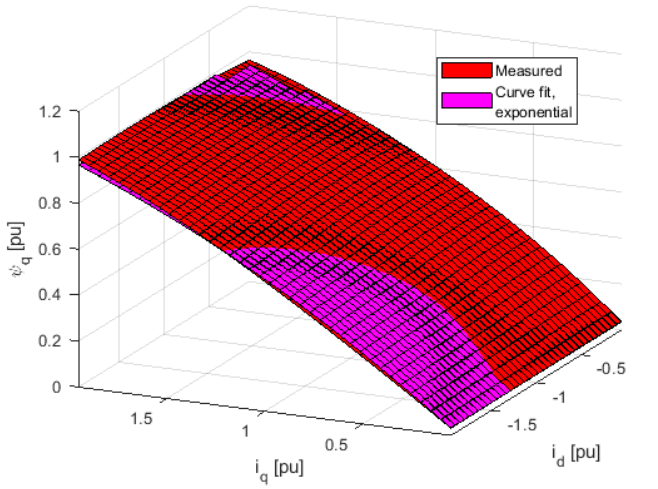


(b)

Fig. 30: The difference between the FE calculated and the model estimated flux linkages by the best polynomial model for 9 measurement points between $i_d \in [-2, 0]$ and $i_q \in [0, 2]$, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

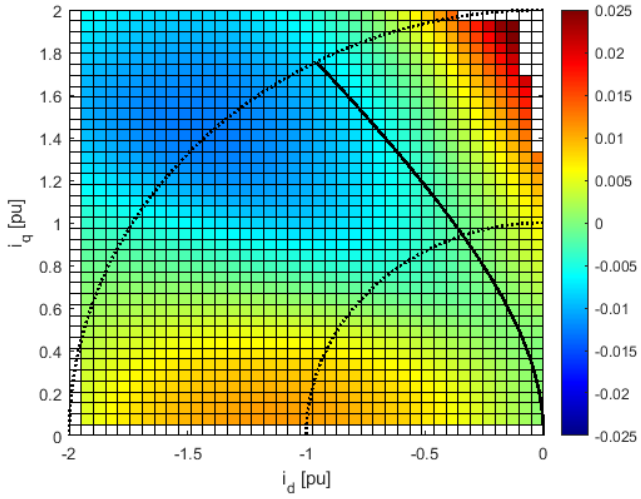


(a)

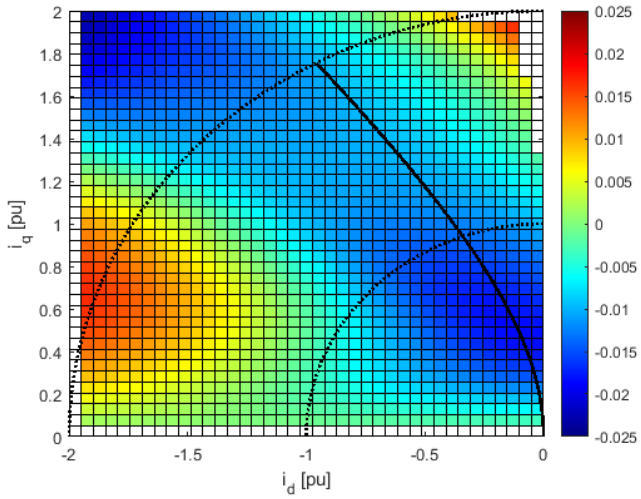


(b)

Fig. 31: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the curve fitted exponential model with coefficients and exponents curve fitted for 9 measurement points in $i_d \in [-2, 0]$ and $i_q \in [0, 2]$.

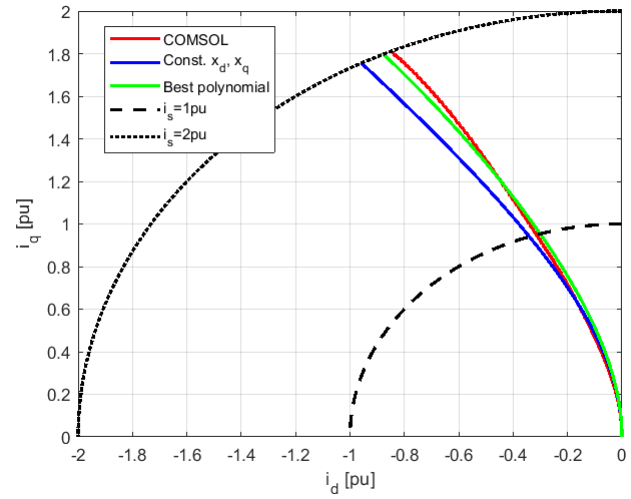


(a)

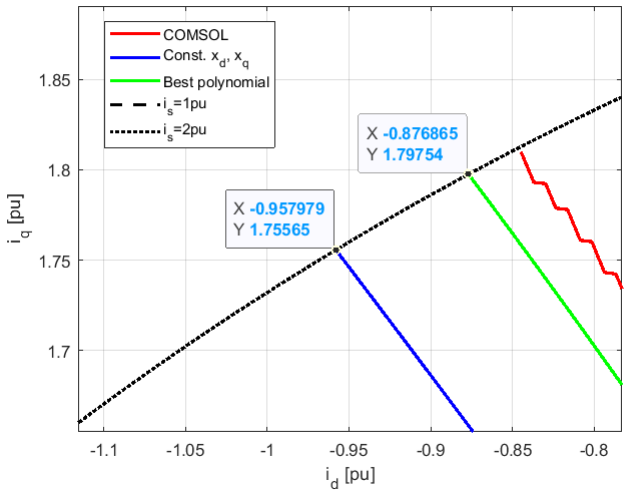


(b)

Fig. 32: The difference between the FE calculated and the model estimated flux linkages by the exponential model for 9 measurement points between $i_d \in [-2, 0]$ and $i_q \in [0, 2]$, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.



(a)



(b)

Fig. 33: (a) The proposed MTPA trajectories either found numerically by COMSOL or analytically by calculating the flux linkages by either the linear model or by the best polynomial model. (b) The proposed operating points to obtain MTPA at a stator current of 2pu, by either calculating the flux linkages by the linear model or by the best polynomial model.

the best polynomial model trajectory is closest to the trajectory calculated by COMSOL, meaning the curve references calculated by the best polynomial model should provide more torque.

At a stator current of 2pu, the proposed current references, meaning combinations of i_d and i_q, corresponded to a load angle of $2.0703rad$ for the linear model, and $2.0246rad$ for the best polynomial model.

B. Analytical MTPA trajectories run through COMSOL

Both the proposed analytically derived current references from Figure 33, were used as input currents to the COMSOL model, such that the torque could be calculated using Arkkio's method. The resulting torques per ampere are given in 34. The proposed MTPA trajectory for the best polynomial model provided more torque per ampere for any given stator current compared to the one of the linear model. At full load, 1pu stator current, the increase in torque when using the best polynomial model was only 0.04% compared to the linear model. For higher currents, an increasing amount of torque per ampere was achieved for the best polynomial model compared to the linear one, and at 2pu stator current the best polynomial model provided 0.33% more torque.

The same results were found when plotting the torque for different load angles at 2pu stator current, as seen in Figure 35. In Figure 35(b) the load angles corresponding to the ones proposed by the MTPA trajectories at 2pu, of $2.0703rad$ and $2.0246rad$ for the linear and best polynomial model respectively, were marked. The relative torque increase when using the best polynomial model was only 0.36% higher, than the torque obtained for the proposed load angle of the linear model.

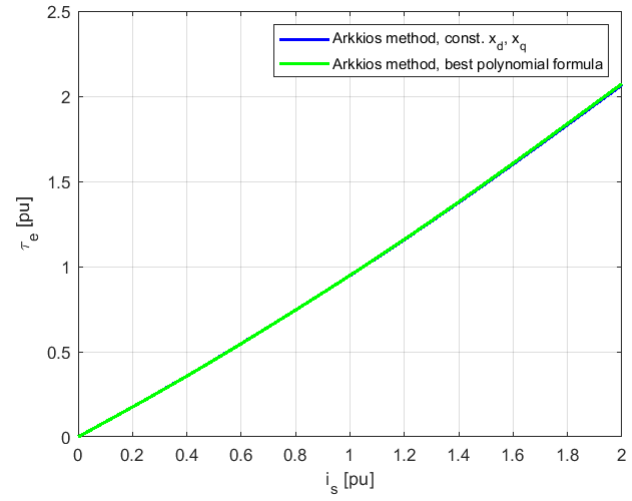
C. Analytical torque calculation versus FE torque calculation

At full load, 1pu stator current, the analytically calculated torque using Equation 12, for the derived current references when using the linear model was 6.35% higher compared to what was calculated by COMSOL for the same current references. At stator currents of 2pu, the torque was analytically calculated to be 12.77% higher than calculated by COMSOL. With higher stator currents, the deviation between the analytically calculated torque by the linear model and the FE calculated torque was increasing, as seen in Figure 36(a).

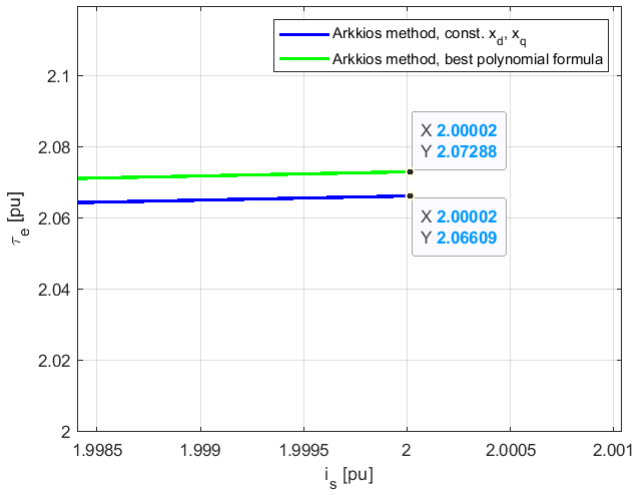
For the best polynomial model, there were no clear correlation between higher stator currents and higher deviations between the analytically torque equation by Equation 40 and the FE calculated torques, as given in Figure 36(b). At 1pu and 2pu stator currents, the deviations between the analytically calculated torque by the best polynomial model to the COMSOL calculated torque were 2.89% and 1.31% respectively.

D. Torque by FE calculated flux linkages and by Arkkio's method

To asses whether the torque deviation between the analytically calculated torque using the best polynomial model and the COMSOL calculated torque was due to the magnetic

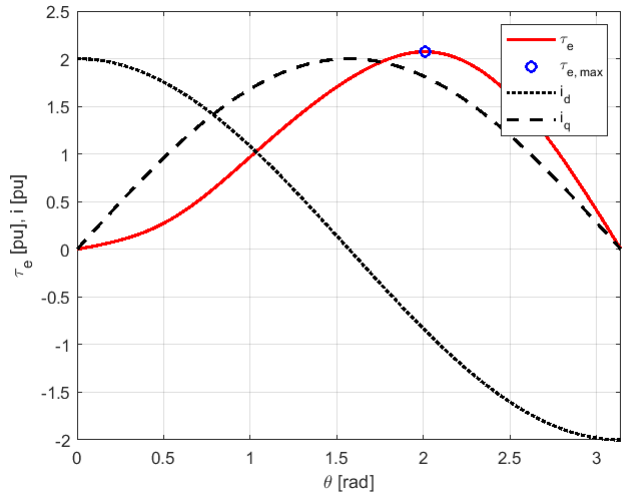


(a)

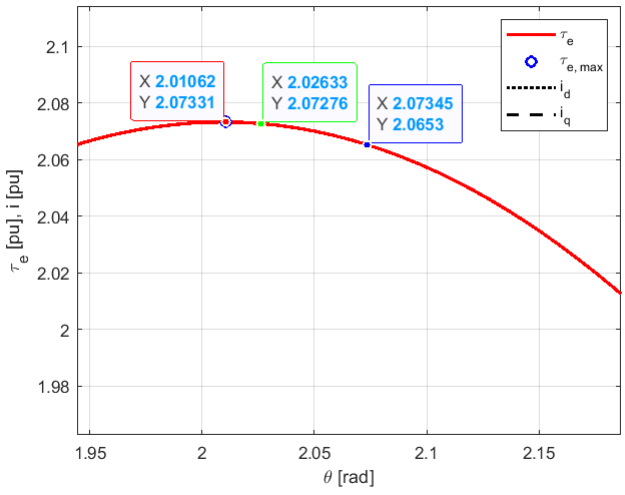


(b)

Fig. 34: (a) Obtained torque in the COMSOL model when applying the proposed combinations of i_d, i_q currents ensuring MTPA, when using either the MTPA trajectory calculated by the linear model or the best polynomial model, as given in Figure 33(a). (b) Obtained torque in the COMSOL model when applying the proposed combinations of i_d, i_q currents ensuring MTPA for a stator current of 2pu, as given in Figure 33(b).



(a)



(b)

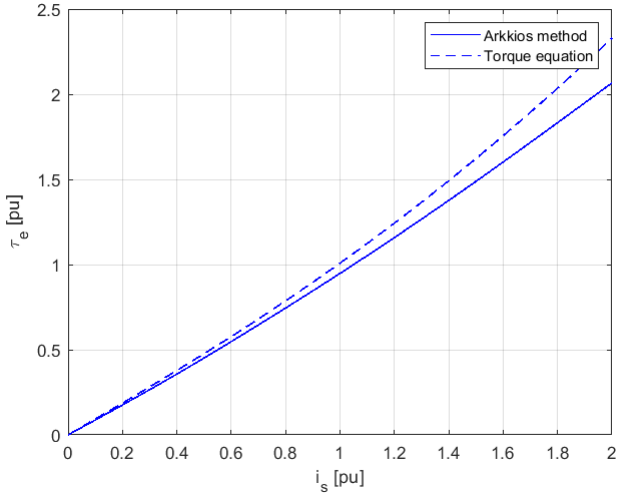
Fig. 35: (a) Torque for different load angles for 2pu stator current. (b) Zoom of (a) showing the numerically found optimal load angle in red, the analytically calculated optimal load angle using the best polynomial model in green, and the analytically calculated optimal load angle using the linear model in blue.

modeling, the flux linkages were directly calculated in the COMSOL model and not estimated by the best polynomial model. Using Equation 7, the analytically calculated torque using the COMSOL calculated flux linkages is given in 37. The analytically calculated torque curve in Figure 37 is close to indistinguishable from the analytically calculated torque in Figure 36(b), where the best polynomial model estimated the flux linkages. This implies that it is not the magnetic modeling of the best polynomial model which creates a deviation between the FE and analytically calculated torque, but rather the analytical torque equation itself.

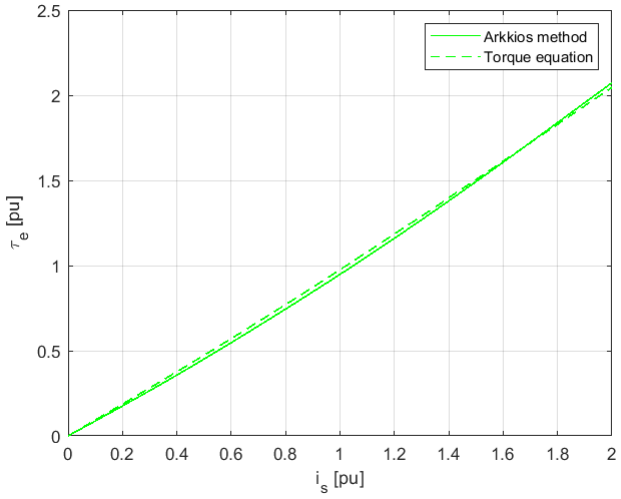
VII. Discussion: Magnetic Modeling

A. Assumptions when building the COMSOL model

When modeling the motor, the turn number, geometry and material properties had to be estimated. In addition, the delta coupled motor was modeled as its star coupled electrical equivalent. Thus, the measured currents and flux linkages in



(a)



(b)

Fig. 36: Torque achieved by Arkkio's method in COMSOL compared to the torque calculated by the analytical torque equation, when using (a) the linear model and (b) the best polynomial model. The same current references were used in COMSOL and the analytical torque equations.

the COMSOL model can not be directly compared to the ones of the physical IPMSM. For instance, at the motor's nameplate, the inductances were reported to be $L_d = 95.3mH$ and $L_q = 206.1mH$, corresponding to per unit constant d- and q-axis reactances of $0.2130pu$ and $0.4607pu$ respectively, which did not equal the reactances derived using the linear model. In addition, no documentation was found stating at which loading conditions the inductances were calculated, making the direct comparison useless. However, although the physical and modeled IPMSMs were not identical, the trends of saturation and cross-coupling are assumed to be comparable, making the analysis in this thesis valid for the

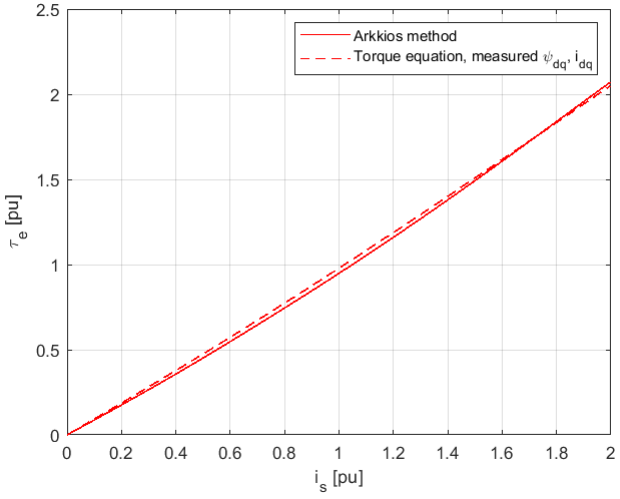


Fig. 37: Torque achieved by Arkkio's method in COMSOL when applying the currents corresponding to the MTPA trajectory of the best polynomial model, compared to the analytically calculated torque by using the calculated d- and q-axis flux linkages and currents directly from COMSOL.

physical motors as well. Nevertheless, this would have to be verified with measurements of the motor at the lab.

B. Flux linkage curves from COMSOL

As was shown in Figure 13, there are both magnetic saturation and cross-coupling phenomena present in the motor as the surface plot of the d-axis flux linkage was changing for different q-axis currents and vice versa. Nevertheless, it might be sufficient for practical purposes to model the flux linkages and applied currents with a linear relation, meaning constant x_d and x_q, neglecting the magnetic saturation and cross-coupling. As was shown when developing the MTPA trajectories where the linear model only provided 0.04% less torque at full load than the best polynomial model.

C. The linear model

In order to calculate ψ_m, x_d and x_q, measurements at only three operation conditions were required. The unsaturated reactances were found using 10% positive currents, as given in [3]. The calculated d-axis flux linkage by x_d would underestimate the actual flux linkages in the machine measured by COMSOL. However, the q-axis flux linkage was overestimated when using the linear model, compared to what was measured by COMSOL. Especially the q-axis flux linkage had a non-linear behavior due to the saturation of the motor iron, leading to a deviation of 17.85% at full load between the estimated and measured flux linkage. As the deviation increased for higher stator currents, it would be advised to calculate x_q for a higher loading than $0.1pu$ q-axis

current. When calculating x_d by 100% negative Id current, the resulting x_d was only 1.31% lower than 10% positive Id current. The x_q for 100% positive Iq would have been 17.06% lower.

D. The simple polynomial model

The simple polynomial model was introduced to find a more accurate constant reactance for the d-axis flux linkage, and model the second order polynomial behaviour of the q-axis flux linkage when the motor went into saturation. By using all 26244 measurement points to curve fit the constant d-axis reactance giving the smallest error to the whole measurement region, the d-axis flux linkage estimation could be greatly improved in the whole operating region compared to the linear magnetic model. At full load of 1pu stator current, the difference from the estimated d-axis flux linkages to the COMSOL ones was reduced from 0.88% for the linear model to 0.39% for the simple polynomial model. For the q-axis flux linkages, the difference was reduced from 17.85% to 3.70%. The difference between the measured and estimated d-axis flux linkage by the simple polynomial model was especially small along the MTPA trajectory, which will ensure a more accurate torque estimation by the control system. The simple polynomial model was estimating the COMSOL computed flux linkages better than the linear magnetic model also when being plotted outside of its curve fitted area. This was also the case when being curve fitted to only 9 measurement points, as seen in Table V, VI and VII. Thus, it is advised to use the simple polynomial model instead of the linear magnetic model, as an accurate simple polynomial model can be developed with very few measurement points. These measurements can be done for a motor at the lab during start-up. The fact that the modeling does not include cross-coupling is more visible for the d-axis than the q-axis. However, the cross-coupling phenomenon is primarily present outside of the typical MTPA operating regions. As cross-coupling was neglected, the reciprocity condition was fulfilled.

E. The best polynomial model

By introducing four more coefficients to the polynomial magnetic modeling, the FE calculated flux linkage curves could be curve fitted accurately by the best polynomial model. At full load and MTPA, the difference between the model estimated d-axis flux linkage to the COMSOL computed one was only 0.23% and only 0.88% for the q-axis, with similar accuracy up to stator currents of 2pu. By the best polynomial model, cross-coupling and saturation were taken into account for both axis flux linkages. However, the reciprocity condition was not fulfilled. Not fulfilling the reciprocity condition means the magnetic model is assumed to not be conservative, such that the inductances, corresponding to the calculated reactances, would be modeled to be able to produce or dissipate energy. By neglecting the reciprocity condition, the curve fitted best polynomial model is only meant to represent the measured reality and not to describe a law of nature correctly. How the best polynomial model would respond to other rotor geometries has not been evaluated, meaning this polynomial version might primarily be optimal for rotors with radial magnetized permanent magnets.

F. The reciprocal polynomial model

For the reciprocal polynomial model, the reciprocity condition was forced to be fulfilled by manipulating the formulas based on the best polynomial model. Although the sum of residuals of the curve fitting was greater for the reciprocal model than the best polynomial model, the reciprocal model had a smaller difference to the measured flux linkages than the best polynomial model within the operating region of 1pu total stator current. Even though the reciprocal model performed better in typical operating regions than the best polynomial model, it was judged that deriving the coefficients by curve fitting each axis flux linkage individually would be more robust when using the models for different motors. If the flux linkages had an even more non-linear behavior for another motor geometry, the reciprocal polynomial model might incorrectly model the flux linkages. Thus, the best polynomial model was used for the further MTPA performance evaluations. The relative performance between the best and reciprocal polynomial model should be studied for different IMPSM designs and experimentally at the lab.

G. The exponential model

The exponential model was also able to accurately curve fit its nine coefficients and exponents to recreate the surface of the FE calculated flux linkages. Having to find the optimal combination of exponents giving the smallest difference to the measured currents and flux linkages, the curve fit had to be recomputed for every exponent combination, meaning the magnetic modeling identification was more computational demanding than of the polynomial functions. However, the exponential model fulfilled the reciprocity condition. Within the typical operating regions, it was performing almost as well as the best polynomial model, with differences to the measured flux linkages of less than 1% for the typical operating regions along the MTPA trajectory.

H. Data extrapolation outside of the curve fitting area

For both the best polynomial model and the exponential model, the models were able to extrapolate accurately outside of their curve fitted areas, compared to the linear model. By [20], it was stated that the performance of the polynomial models would deteriorate if they were to extrapolate values outside of their curve fitted area. However, when using the

best polynomial model, it had a smaller difference to the FE calculated flux linkage in the d-axis flux linkage compared to the exponential model. For the q-axis flux linkage, the exponential and best polynomial model performed relatively equally. In conclusion, both models were able to extrapolate outside of their curve fitted areas. However, if the operating regions of the motor would be known, the models should be curve fitted to the whole known operating region.

It should be noted that for both the best polynomial model and the exponential model, the difference had decreased between the model estimated and FE measured flux linkages within the curve fitting area, compared to when the curve fitting was done to the whole operating area up to 2pu. This means the accuracy of the curve fittings can possibly be improved by calculating different model coefficients for different operating areas. However, by developing different coefficients for different regions, it is important that between these regions, the derivative is continuous.

I. Performance of curve fitting with only 9 measurement points

For 9 measurement points, all the models were able to curve fit such that the models performed better than the linear magnetic model. The simple polynomial model had the largest relative deviations to the FE calculated flux linkages, but only 0.39% for the d-axis and 3.70% for the q-axis at full load and MTPA operation. As the models had smaller deviations to the measured flux linkages when curve fitting using 26244 measurement points compared to 9, it is advised to include as many measurement points as possible in order to achieve an as detailed model as possible. However, with only 9 measurement points, the models performed a lot better than the linear model which requires 3 measurement points to develop x_d and x_q, thus developing magnetic models by curve fitting is advised compared to using the linear model. As only 9 measurement points are required, this enables plug-and-play methods, meaning an undescribed IPMSM could be efficiently magnetically modeled based on few measurement points during start-up based on experiments at the lab.

VIII. Discussion: Control improvement

A. Obtained torque for analytical MTPA trajectories

As the current references to achieve MTPA operation were calculated using the torque equation, it was assumed that using a more accurate magnetic model than the linear one would increase the torque per ampere in the IPMSM. Using the best polynomial formula as the magnetic model, the developed MTPA trajectory was close to what was simulated to be the best one in COMSOL. As the linear model developed a MTPA trajectory with more negative d-axis current than COMSOL calculated to ensure maximum torque, the best polynomial model was assumed to give a higher torque for the same ampere compared to the linear model.

However, when the proposed d- and q-axis current combinations providing maximum torque per ampere were simulated in the COMSOL model, the resulting torque for the current references of the best polynomial model was negligible higher than the the torque obtained by the current references of the linear model. At full load, the current references of the best polynomial model only provided 0.04% higher torque relative to the torque using the linear model. When increasing the stator current beyond full load up to 2pu, the current references derived from the best polynomial model would provide 0.33% higher torque than if the linear model was used. Thus, despite the motor being deep into saturation, which the linear model neglected, the derived MTPA trajectory of the linear model provided efficient torque control. As the linear model underestimated the d-axis flux linkage and overestimated the q-axis flux linkage along the MTPA trajectory, as shown in Figure 16, it might be that the combination of these effects provided a satisfactory MTPA trajectory. It is still advised to use the best polynomial model to calculate the MTPA trajectory. The linear model might not give an accurate trajectory for motors of different geometries and material properties. However, it is computationally more demanding to derive the MTPA trajectory from the best polynomial model. In addition, the MTPA trajectory could be explicitly calculated analytically using the linear model, but the MTPA trajectory of the best polynomial model had to be solved numerically, which might introduce convergence issues.

B. Improved torque estimation

A benefit of modeling the flux linkages using the best polynomial model was that the analytically calculated torque was much closer to what was measured in COMSOL than the linear model. For the same current references at full load, the best polynomial model estimated a torque that was 2.89% higher than the actual torque measured by Arkkio's method, while the linear model calculated a torque that was 6.35% higher. This was due to the fact that as the linear model neglected the magnetic saturation, the q-axis flux linkage was highly overestimated in the machine, meaning the torque also became overestimated. Thus, the more the motor iron became saturated, the more the linear model overestimated the torque compared to the FE calculated one. The best polynomial model, which took the magnetic saturation and cross-coupling into account, did not overestimate the torque as the motor went into saturation. If the motor was to be torque controlled without a speed controller, the linear model would, based on the applied currents, estimate a higher torque in the motor than what would actually be the case. So if there is no speed controller in the motor control system, the control system might overestimate the torque in the machine and think it

has reached its reference, while the actual torque in the machine is lower. Due to this, the motor might decelerate, depending on the load. Using the best polynomial model, a more accurate torque estimation closer to what would be actually experienced by the motor would be achieved. However, as there were still deviations between the model estimated and FE calculated torque using the best polynomial model, a speed controller should be integrated in the motor control, such that the torque reference could be chosen based on the torque demand at the load.

C. Torque equation evaluation

It was assumed that the analytical torque equation was not accurately calculating the torque in the machine due to two observations. Firstly, even though the best polynomial model would accurately model the flux linkages, a deviation was observed between the calculated MTPA trajectory using COMSOL and the analytically derived MTPA trajectory using the best polynomial model. Secondly, when comparing the analytically calculated torque and the torque calculated by Arkkio's method in COMSOL, the analytical equation provided a higher torque. Thus, to evaluate the accuracy of the analytical torque equation, the flux linkages and applied currents were both measured in COMSOL. When calculating the torque based on the measured flux linkages and currents, the torque did not equal the calculated torque by Arkkio's method. The analytically calculated torque calculated by the measured flux linkages was identical to the torque calculated when using the best polynomial model, as seen from Figure 36(b) and Figure 37. This implies that it was not the magnetic modeling that was the reason for the difference in torque by the analytical torque equation and Arkkio's method, but the torque equation itself.

As mentioned in Section II-A, the per-unit model of the motor assumes that the magnetic flux density in the air gap is sinusoidally distributed. However, in Figure 38, the air gap flux density for a sector of the IPMSM was plotted, showing that the air gap flux density was closer to being square-shaped, containing third and sixth order harmonics. When the torque was calculated by Arkkio's method, the spatial harmonics were included as the torque calculation was done by Maxwell's stress tensor method. It is thus assumed that the analytical torque equation should be expanded to include the other higher-order spatial harmonics than just the fundamental to calculate the same torque as Arkkio's method. For applications where the IPMSM needs to provide high accuracy, this is advised. However, for less demanding applications, the motor control should include a speed controller, such that the motor is always provided with the necessary torque. Alternatively, the design of the motor should be made such that the magnetic flux density in the air gap would be sinusoidal. For instance by making the rotor

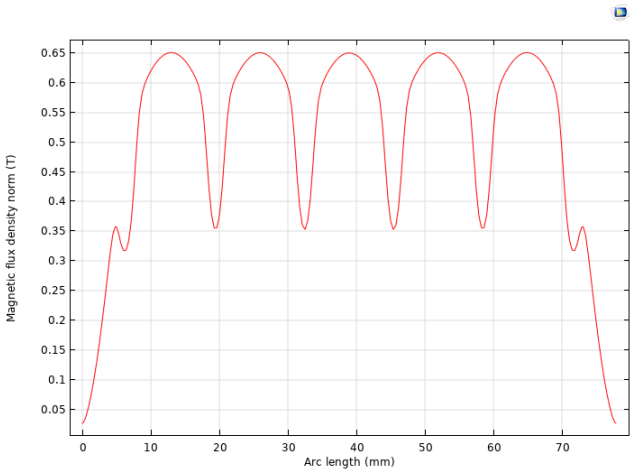


Fig. 38: The air gap flux density of the IPMSM due to the permanent magnet, at no-load.

geometry more ellipse shaped than circular for each sector.

Thus, the control system of the IPMSM could be improved by three measures, 1) using a detailed magnetic model, 2) extending the torque equation to take into account the spatial harmonics of the magnetic flux density in the air gap and 3) redesigning the motor geometry or winding scheme to provide a sinusoidal shaped magnetic flux density in the air gap.

IX. Conclusion

In this thesis, several explicit formulas with different degrees of complexity have been presented to model the flux linkages in an IPMSM, focusing on the stator current influence. Using a FE model approximated to a commercial motor, d- and q-axis flux linkages have been calculated for applied d- and q-axis currents. Magnetic saturation and cross-coupling phenomena have been shown to exist in the motor. It has also been shown that the non-linear magnetic behaviors are not efficiently modeled using a linear magnetic model, assuming constant x_d, x_q, especially when calculating the q-axis flux linkage. At full load, operating with MTPA control, the relative difference between the FE calculated and the linearly calculated q-axis flux linkage was 17.85%. The magnetic models could be improved by introducing non-linear models using curve fitting of the FE calculated data.

All of these models were based on curve fitting of the flux linkages with variable d-axis currents ranging from $-2pu$ to $0pu$, and q-axis currents ranging from $0pu$ to $2pu$. For the proposed *simple* polynomial model, the q-axis flux linkage was modeled to have a second-order relation with the applied q-axis current. By this measure, the difference between the FE calculated and model estimated q-axis flux linkage was reduced to 3.70% at MTPA operation and a total stator current of $1pu$, meaning full load. Increasing the number

of coefficients, as for the *reciprocal* and *best* polynomial model with 6 and 7 coefficients respectively, only gave minor deviations to the FE calculated flux linkages with under 1% at full load and MTPA control. Both the *reciprocal* and the *best* polynomial models take into account the magnetic saturation and cross-coupling, but only the *reciprocal* model takes the reciprocity condition into account. Finally, a pre-described model from the literature taking the reciprocity condition into account was studied and called the *exponential* model. The model consisted of 5 coefficients and 4 exponents to be curve fitted and was thus computationally more demanding to solve when using the linear least square method than the polynomial models. The *exponential* model was less accurate than the *best* polynomial model and the *reciprocal* polynomial model within the typical operating regions. However, the *exponential* model was still accurate enough to result in relative deviations from the FE calculated flux linkages of less than 1%. At full load and MTPA operation, the relative difference between the FE measured and model estimated d-axis flux linkage was only 0.16% and 0.92% for the q-axis flux linkage.

All curve fitted models could extrapolate outside of the area they were curve fitted to, but the polynomial models were more accurate than the exponential one. Curve fitting the coefficients to only 9 measurement points proved satisfactory for all models, enabling rapid magnetic model identification during the commissioning of new undescribed motors.

Based on the magnetic modeling, the thesis has shown how the IPMSM might achieve more torque per ampere as well as improved analytical torque calculation when using the *best* polynomial model instead of the *linear* model. Nevertheless, when calculating the current references at full load to ensure MTPA operation using the *best* polynomial model, the resulting torque was only 0.04% higher compared to when the current references were calculated using the *linear* model. The benefit of calculating the current references based on the *best* polynomial model increased with higher stator currents, as the magnetic saturation in the motor would increase as well. However, if the purpose of the magnetic modeling is to ensure MTPA operation, the *linear* model is judged to be accurate enough. For the same current references at full load, the analytically calculated torque using the *best* polynomial model was 2.89% higher than what was calculated in the COMSOL model by Arkkio's method. As the magnetic modeling of the *best* polynomial model was very accurate, the analytical torque deviation from the torque calculated by Arkkio's method was argued to be due to the neglect of the spatial harmonics of the magnetic flux density in the air gap, which should be included in the analytical torque equation for applications demanding precise torque control.

X. Future work

The following tasks were not covered, but are judged to be natural continuations of the thesis.

1) Verify the conclusions done for the proposed models by lab investigation.
2) Investigate if the magnetic models are as accurate for other motor geometries.
3) Improve the COMSOL model by collecting the data of the motor materials and windings.
4) Develop a delta connected COMSOL model and map the flux linkages based on the measured voltages during a dynamic analysis, as done at the lab.
5) Connect the COMSOL model to the control system in Simulink using the COMSOL extension Livelink for Simulink, to simulate the potential motor control improvement by including the magnetic modeling in the control structure.
6) Design a motor in COMSOL providing a more sinusoidal air gap flux density and investigate if the torque estimation would be improved.

XI. Acknowledgements

I would like to express gratitude to Robert Nilssen, Roy Nilsen and Aravinda Perera for supervising this project and for providing valuable insight essential for the completion of this thesis. I would also like to thank my family for the continuous support. In addition, I am grateful for the help I have received from Magnus N. Malmquist regarding the linear regression development and from Runar Mellerud regarding the revision of the thesis text. Finally, I would like to thank my flat mates and other friends for making this period fun and memorable, despite the thesis being written during the covid-19 pandemic.

References

[1] Aravinda Perera and Roy Nilsen. A sensorless control method for ipmsm with an open-loop predictor for online parameter identification. In *2020 23rd International Conference on Electrical Machines and Systems (ICEMS)*, pages 1983–1988, 2020.

[2] Thomas M. Jahns, Gerald B. Kliman, and Thomas W. Neumann. Interior permanent-magnet synchronous motors for adjustable-speed drives. *IEEE Transactions on Industry Applications*, IA-22(4):738–747, 1986.

[3] Nicola Bianchi. *Electrical Machine Analysis Using Finite Elements*. 01 2005.

[4] Hafiz Asad Ali Awan, Zhanfeng Song, Seppo E. Saarakkala, and Marko Hinkkanen. Optimal torque control of saturated synchronous motors: Plug-and-play method. *IEEE Transactions on Industry Applications*, 54(6):6110–6120, 2018.

[5] Marko Hinkkanen, Seppo E. Saarakkala, Hafiz Asad Ali Awan, Eemeli Mölsä, and Toni Tuovinen. Observers for sensorless synchronous motor drives: Framework for design and analysis. *IEEE Transactions on Industry Applications*, 54(6):6090–6100, 2018.

[6] Francois J. W. Barnard, Wikus T. Villet, and Maarten J. Kamper. Hybrid active-flux and arbitrary injection position sensorless control of reluctance synchronous machines. In *2014 International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, pages 1146–1151, 2014.

[7] Christian Rivera, Javi Poza, G. Ugalde, and Gaizka Almandoz. Field weakening characteristics computed with fem-coupled algorithms for brushless ac motors. *Energies*, 11:1288, 05 2018.

[8] Mohammad Hossain Mohammadi and David Alister Lowther. A computational study of efficiency map calculation for synchronous ac motor drives including cross-coupling and saturation effects. *IEEE Transactions on Magnetics*, 53(6):1–4, 2017.

[9] Hugo W. de Kock, Arnold J. Rix, and Maarten J. Kamper. Optimal torque control of synchronous machines based on finite-element analysis. *IEEE Transactions on Industrial Electronics*, 57(1):413–419, 2010.

[10] Gianmario Pellegrino, Barbara Boazzo, and Thomas M. Jahns. Magnetic model self-identification for pm synchronous machine drives. *IEEE Transactions on Industry Applications*, 51(3):2246–2254, 2015.

[11] K.M. Rahman and S. Hiti. Identification of machine parameters of a synchronous motor. In *38th IAS Annual Meeting on Conference Record of the Industry Applications Conference, 2003.*, volume 1, pages 409–415 vol.1, 2003.

[12] Eric Armando, Radu Iustin Bojoi, Paolo Guglielmi, Gianmario Pellegrino, and Michele Pastorelli. Experimental identification of the magnetic model of synchronous machines. *IEEE Transactions on Industry Applications*, 49(5):2116–2125, 2013.

[13] Nicola Bedetti, Sandro Calligaro, and Roberto Petrella. Stand-still self-identification of flux characteristics for synchronous reluctance machines using novel saturation approximating function and multiple linear regression. *IEEE Transactions on Industry Applications*, 52(4):3083–3092, 2016.

[14] B. Stumberger, G. Stumberger, D. Dolinar, A. Hamler, and M. Trlep. Evaluation of saturation and cross-magnetization effects in interior permanent-magnet synchronous motor. *IEEE Transactions on Industry Applications*, 39(5):1264–1271, 2003.

[15] Hafiz Asad Ali Awan, Seppo E. Saarakkala, and Marko Hinkkanen. Flux-linkage-based current control of saturated synchronous motors. *IEEE Transactions on Industry Applications*, 55(5):4762–4769, 2019.

[16] A. Kilthau and J.M. Pacas. Appropriate models for the control of the synchronous reluctance machine. In *Conference Record of the 2002 IEEE Industry Applications Conference. 37th IAS Annual Meeting (Cat. No.02CH37344)*, volume 4, pages 2289–2295 vol.4, 2002.

[17] Shu Yamamoto, Takahiro Ara, and Kouki Matsuse. A method to calculate transient characteristics of synchronous reluctance motors considering iron loss and cross-magnetic saturation. *IEEE Transactions on Industry Applications*, 43(1):47–56, 2007.

[18] Roy Nilsen. *TET4120 Electric Drives*. 2018.

[19] Waqar Khan. Torque Maximizing and Flux Weakening Control of Synchronous Machines. Master's thesis, Aalto University. School of Electrical Engineering, 2016.

[20] Zengcai Qu, Toni Tuovinen, and Marko Hinkkanen. Inclusion of magnetic saturation in dynamic models of synchronous reluctance motors. In *2012 XXth International Conference on Electrical Machines*, pages 994–1000, 2012.

[21] A. Vagati, M. Pastorelli, F. Scapino, and G. Franceschini. Impact of cross saturation in synchronous reluctance motors of the transverse-laminated type. *IEEE Transactions on Industry Applications*, 36(4):1039–1046, 2000.

[22] Kai Sun and Qifang Shu. Field weakening operation control strategies of interior permanent magnet synchronous motor for electric vehicles. In *Proceedings of the 30th Chinese Control Conference*, pages 3640–3643, 2011.

[23] Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, and Keying Ye. *Probability & statistics for engineers & scientists, ninth edition*. 2016. pg. 460.

[24] Øyvind S. Klyve. Additive manufacturing of electrical machines by the use of fdm and magnetic pla, 2020.

[25] COMSOL Multiphysics. How to analyze an induction motor: A team benchmark model, 2016. https://www.comsol.com/blogs/how-to-analyze-an-induction-motor-a-team-benchmark-model/.

[26] J. Ravi Kumar Mr. and Dr. Banakara, Basavaraja. Finite Element Analysis in the Estimation of Air-Gap Torque and Surface Temperature of Induction Machine. In *Materials Science and Engineering Conference Series*, volume 225 of *Materials Science and Engineering Conference Series*, page 012116, August 2017.

APPENDIX A
COMSOL MODELLING

A. COMSOL modeling

These modeling details were originally written for the *Specialization Project* prior to the thesis [24], and is regiven here with a few alterations as the master thesis is not a continuation of the specialization project.

1) Geometry: Having opened and measured a comercially available IPMSM, a 2D model of the motor was built based on its measured geometry. The model of the motor was then cut into a sector of 1/6 of the whole motor geometry, as this sector would be antiperiodisticly symmetrical for 1/6 of the whole geometry. In order to be able to use Arkkio's method when calculating the torque, two equally wide air gap bands were defined between the rotor and the stator. The 2D geometry of the modelled sector can be found in Figure 2. The length of the motor was set to be 6 times longer than the measured length, to accurately calculate the induced voltage, flux linkage and torque, despite only having modelled one sector, meaning one sixth of the motor. Key geometry data is given in Table VIII.

2) Materials: The magnets being assumed to be of the type N50, were assigned the default COMSOL material *N50 (Sintered NdFeB)*. The rotor and stator materials were set to have the built-in material *Silicon Steel NGO 35PN300*, the coils *Copper*, the plate between the magnet and rotor was set to *Steel AISI 4340* and the rest of the domains *Air*. The materials were chosen to best replicate the unknown materials of the actual motor. As the silicon steel sheets of the stator and rotor were laminated, eddy currents in these areas were neglected. This was achieved by setting the conductivity of the silicon steel to 1e-6S.

3) Physics: The motor physics were modeled using the physics module *Rotating Machinery, Magnetic*. Antiperiodistic boundary conditions were defined at each sector border, using *Periodic Condition*. The *Sector Symmetry* was defined at the boundary between the two air gap bands, meaning the border between the rotating and stationary parts, and the symmetry was set to antiperiodistic with 6 sectors.

In the stator domain, the function *Ampère's Law* was defined to use the *B-H Curve* magnetization model. Meaning the relative permeability and saturation of the stator and rotor steel would be included in the model.

The magnets were also defined using *Ampères Law*, and by selecting the magnetization model *Remanence flux density*. The magnet was defined to have its remanence flux direction as 1 along the r-axis.

For each pair of slots, assumed to be 102 number of turns for each slot, with a round wire diameter of 0.75[mm] and a conductivity of 5e7[S/m]. The input currents were given on amplitude value with ω as the angular frequency. The three phases, a, b, c were defined in COMSOL as given in Equation 14, and the winding scheme is given in Figure 2.

4) Mesh: The rotor, the magnet, the air domain between the magnet and rotor and one of the air gap bands were in *Moving Mesh* defined as *Rotating Domains*, with the chosen rpm selected as *Constant revolutions per time*. For the stationary analysis these definitions were not necessary, but the model was prepared to be able to run in a dynamic analysis.

The mesh at each side of the sector, and at each air domain boundary, were defined using *Edge* and *Copy Edge* to define an equal number of nodes at each side of the sector. All the domains of the sector were chosen to have an *Extremely Fine* mesh. Having such a dense mesh would normally be a great exaggeration, but the model did not converge otherwise when being pushed deep into saturation. As the Parametric sweeps were not too time consuming to solve, the mesh was left dense. Using the function *Distribution* and *Mapped* and selecting the three boundaries in the air gap bands, the number of square mesh elements could be set to 200.

5) Study: For the analysis, the load angle between the rotor magnet and stator currents were changed by keeping the rotor in a constant position in relation to the stator while changing the angles of the stator currents. By this method, there would be no rotating parts in the machine such that the study could be performed by a *Parametric Sweep* of *Stationary* study methods. The sweeps were done by either defining specific I_d and I_q currents, or defining a stator current amplitude, I_s, and changing the angle offsets of the currents, θ, as explained in Section III.

6) Results: After solving the model, the data had to be post-processed. The results of interest for this model were the air gap flux density, the flux linkages, the magnetic saturation of the stator and rotor and the torque. The air gap flux density in the air gap was found by selecting the boundary between the two air gap bands using *Line graph* in a *1D Plot Group* function, and by inserting the expression *(rmm.Bx*X+rmm.By*Y)/sqrt(X∧2+Y∧2)*. The magnetic saturation in the motor was visualized by plotting the relative permeability in a *Surface* plot for a *2D Plot Group*, using the expression *rmm.normB/(mu0_const*rmm.normH)*. The torque was calculated using Arkkio's method, a variation of Maxwell's stress tensor method, given in Equation 43 [25] [26]. Meaning the torque was calculated by integrating over the radius, r, and the magnetic flux density in radial direction, B_r, and tangential direction, B_θ.

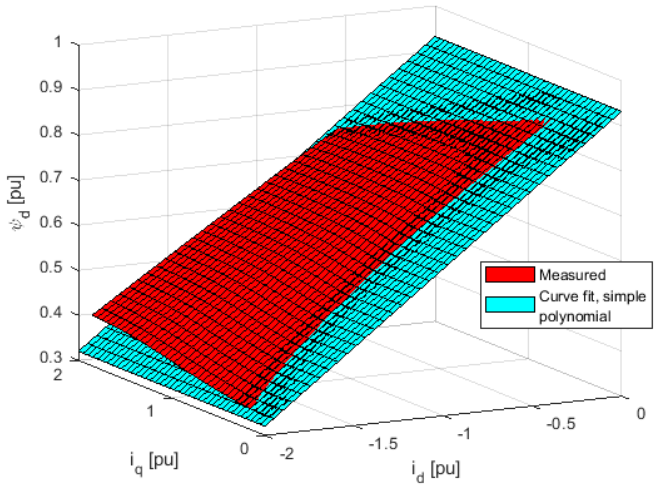
TABLE VIII: Key data of the modelled motor in COMSOL.

Name	Expression	Value	Description
Ls	80.8[mm]	0.0808 m	Axial length stator
Dso	220[mm]	0.22 m	Outer stator diameter
Dsi	150[mm]	0.15 m	Inner stator diameter
Dri	47.5 [mm]	0.0475 m	Inner rotor diameter
hsu	20[mm]	0.02 m	Slot height stator
hPM	5[mm]	0.005 m	Height PM
wPM	60[mm]	0.06 m	Width PM
Dro	147.5[mm]	0.1475 m	Outer rotor diameter
Thh	1.5[mm]	0.0015 m	Tooth head height
Tws	6.2[mm]	0.0062 m	Tooth width stem
Twh	10[mm]	0.01 m	Tooth width head
slots	36	36	Slots/teeth
mag_slot_lower	Dri/2+32[mm]	0.05575 m	Lower part of mag slot
mag_slot_height	6[mm]	0.006 m	Magnet slot height
D_slot_circ	4.5[mm]	0.0045 m	Circle in magnet slot
air_gap	(Dsi-Dro)/2	0.00125 m	Air gap
rpm	1000 [1/min]	16.667 1/s	RPM
p	6	6	Poles
f	rpm*p/2	50 1/s	Frequency
w	2*pi*f	314.16 1/s	Angular frequency

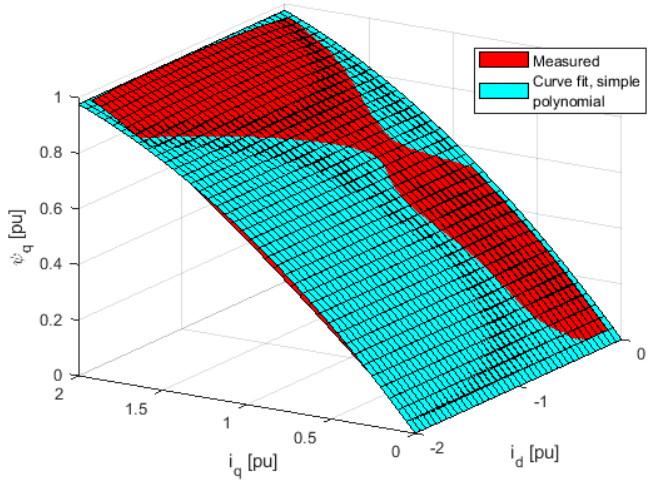
$$T_e = \frac{l}{\mu_0(r_{out} - r_{in})} \int_S B_r B_\theta r dS \tag{43}$$

Where l was the length of the machine, while r_{in} was the inner radius of the air gap band closest to the center, and r_{out} was the outer radius of the outer air gap band. In COMSOL, an *Integration* function was defined for the two air gap bands under *Definitions*. Then by defining T_e in *Variables* with the expression *(p*L/(mu0_const (r_out-r_in)))*intop_torque((X*rmm.BX+Y*rmm.BY)*(Y*rmm.BX-X*rmm.BY)/sqrt(X∧2+Y∧2))*, the torque could be calculated. T_e was finally plotted in a *1D Plot Group* as a *Global* variable, and the average torque was calculated in a *Global Evaluation* in *Derived Values*. The flux linkages, or *Coil concatenated flux* as called in COMSOL, were given for each phase, in this case phase a, as *rmm.PhiCoil_a*. The d- and q-axis currents and flux linkages were derived using Equation 15 and 16 respectively.

APPENDIX B
SURFACE PLOTS

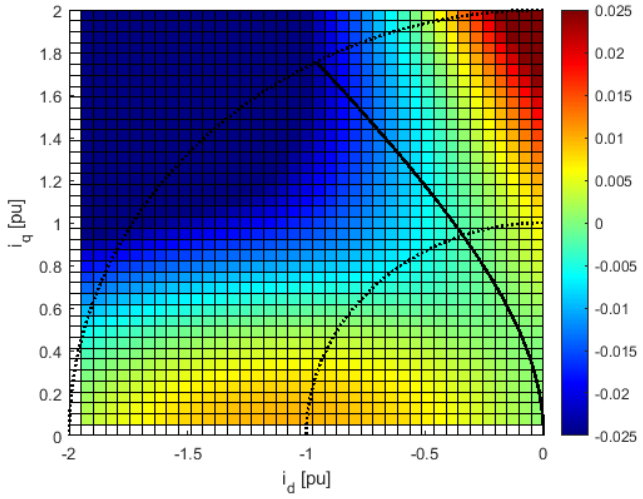


(a)

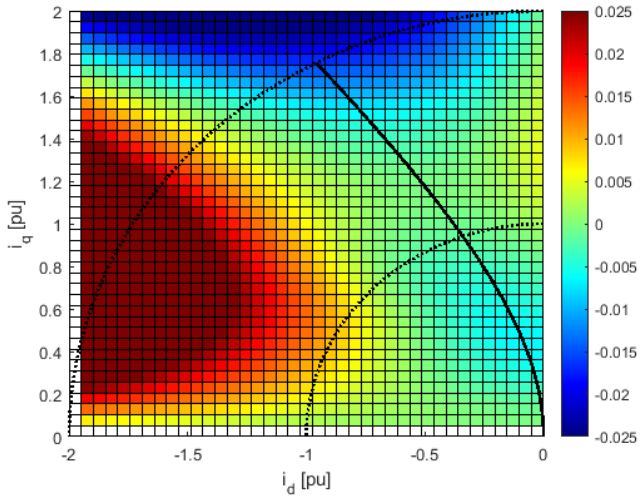


(b)

Fig. 39: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the simple polynomial model with coefficients curve fitted for 6561 measurement points in $i_d \in [-1, 0]$ and $i_q \in [0, 1]$.

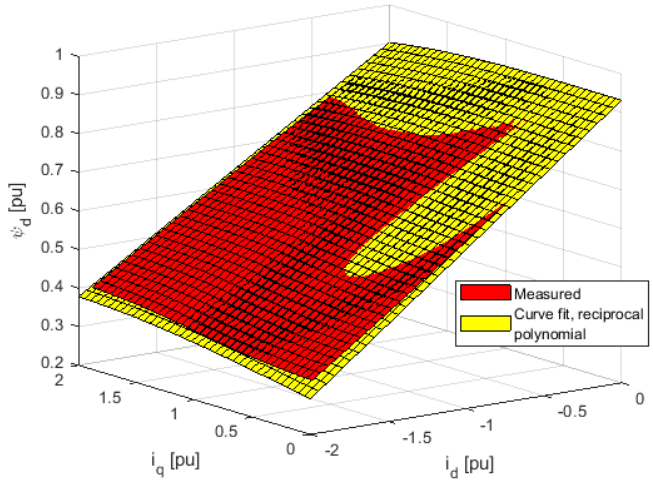


(a)

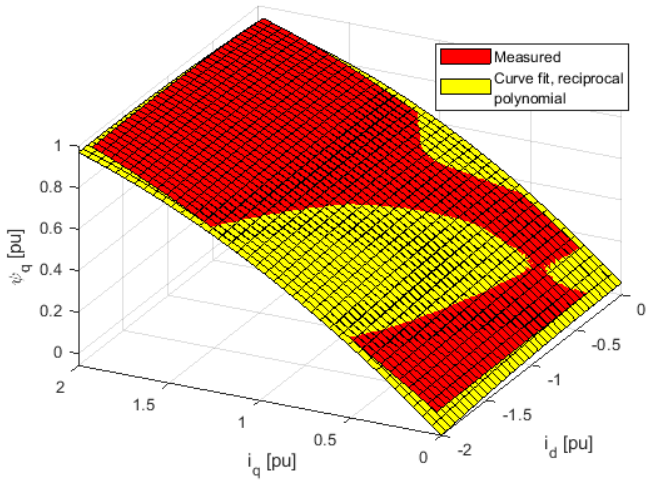


(b)

Fig. 40: The difference between the FE calculated and the model estimated flux linkages by the simple polynomial model for 6561 measurement points between $i_d \in [-1, 0]$ and $i_q \in [0, 1]$, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

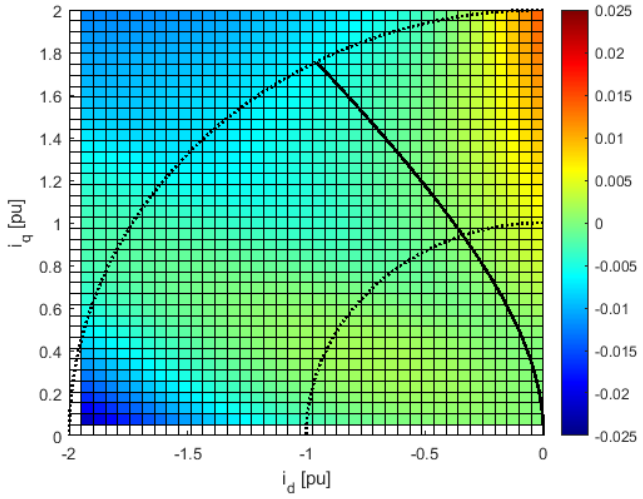


(a)

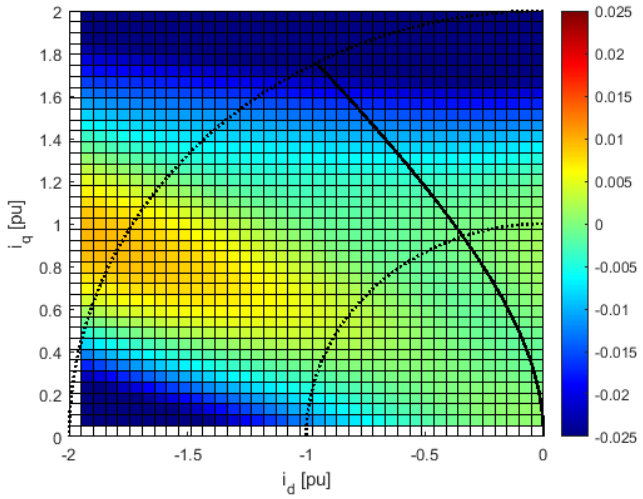


(b)

Fig. 41: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the reciprocal polynomial model with coefficients curve fitted for 6561 measurement points in $i_d \in [-1, 0]$ and $i_q \in [0, 1]$.

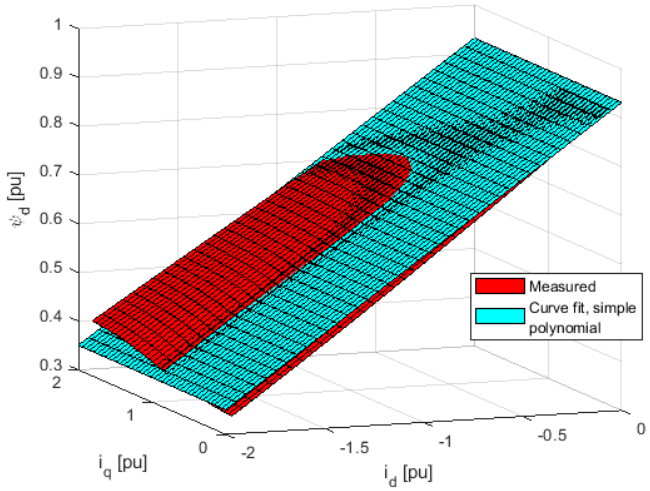


(a)

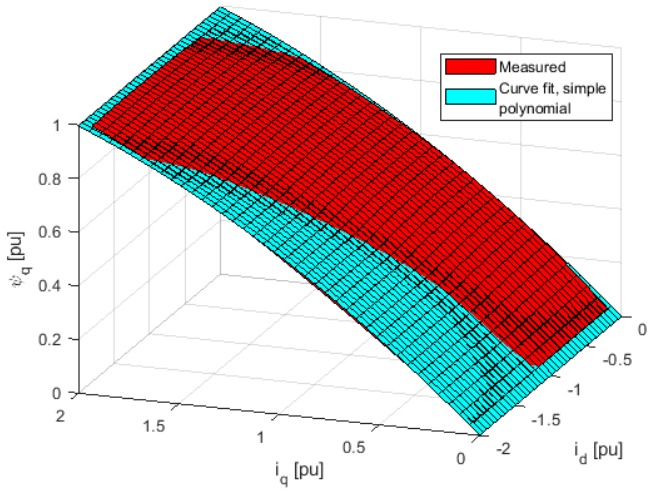


(b)

Fig. 42: The difference between the FE calculated and the model estimated flux linkages by the reciprocal polynomial model for 6561 measurement points between $i_d \in [-1, 0]$ and $i_q \in [0, 1]$, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

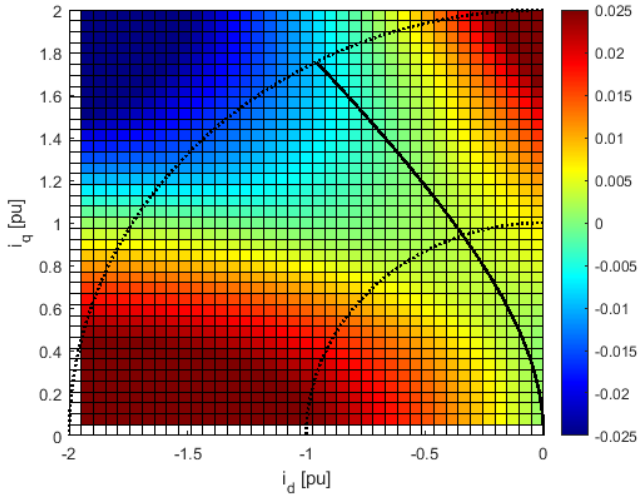


(a)

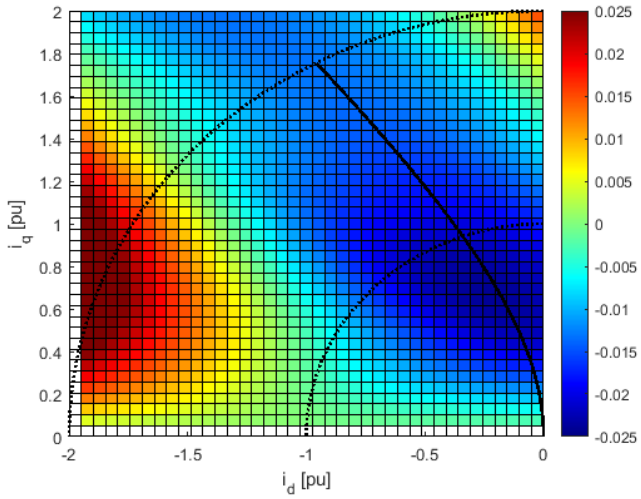


(b)

Fig. 43: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the simple polynomial model with coefficients curve fitted for 9 measurement points in $i_d \in [-2, 0]$ and $i_q \in [0, 2]$.

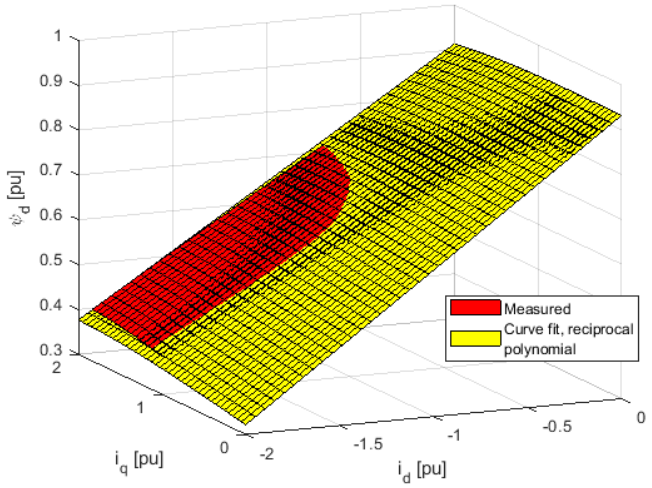


(a)

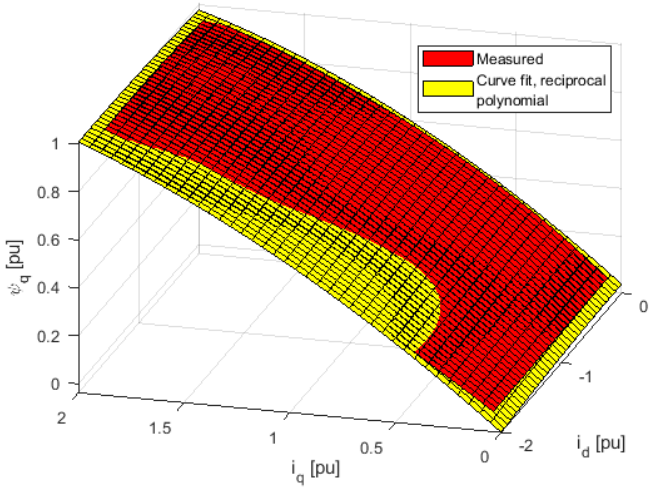


(b)

Fig. 44: The difference between the FE calculated and the model estimated flux linkages by the simple polynomial model for 9 measurement points between $i_d \in [-2, 0]$ and $i_q \in [0, 2]$, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

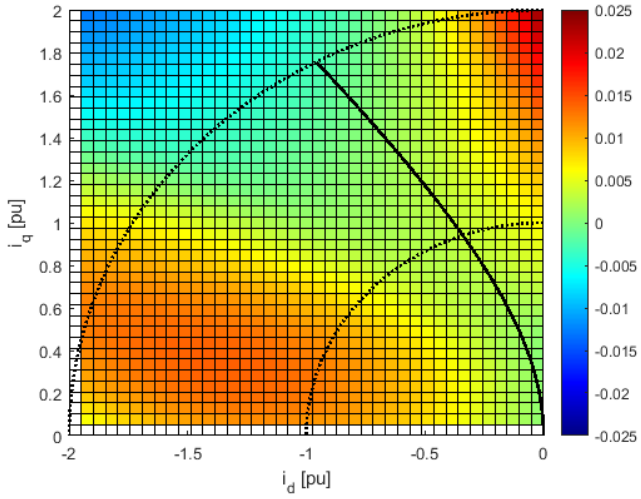


(a)

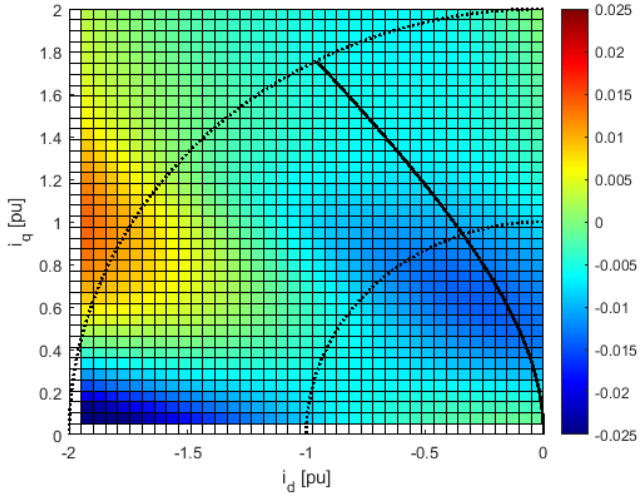


(b)

Fig. 45: The FE calculated (a) ψ_d and (b) ψ_q in COMSOL, plotted with the calculated ψ_d and ψ_q when using the reciprocal polynomial model with coefficients curve fitted for 9 measurement points in $i_d \in [-2, 0]$ and $i_q \in [0, 2]$.



(a)



(b)

Fig. 46: The difference between the FE calculated and the model estimated flux linkages by the reciprocal polynomial model for 9 measurement points between $i_d \in [-2, 0]$ and $i_q \in [0, 2]$, for (a) ψ_d and (b) ψ_q. The MTPA trajectory and plots of the stator currents of 1pu and 2pu from Figure 6 are marked to emphasize the operating region of the motor.

APPENDIX C
THE MATLAB CODES

A. *Initializing curve fitting*

```
%CURVE FITTING ...
    MAIN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load_data=csvread('New_Ld_Lq_2_Idq.csv',5,0); ...
    %Main measurement
load_datat=csvread('New_Ld_Lq_9points_230521.csv'
,5,0); %Few meas
%Real values
I_d=load_data(:,5);
I_q=load_data(:,6);
Psi_d=load_data(:,7);
Psi_q=load_data(:,8);

Psi_m=1.6781; %NEW

%Calculated Ld Lq
Ld=143.14e-3; %NEW, 10% of In
Lq=327.64e-3; %NEW, 10% of In
%Ld=123.58e-3; %NEW 100% of pos In, %141.27 ...
    if 100% neg In
%Ld=141.27e-3;
%Lq=271.73e-3; %NEW 100% of pos In, %value ...
    for new xq in plots

%% Motorparameters
p=3; %pair of poles
U_line= 400; %V
I_line=4.93; %A
fn=50; %Hz
wn=2*pi*fn;
Xd=Ld*wn;
Xq=Lq*wn;
Rs=4.4;

%Base values
U_base=U_line*sqrt(2); %Peak Voltage per ...
    phase!!!!!!!!!!
I_base=I_line*sqrt(2)/sqrt(3); %Peak current
Psi_base= U_base/(wn);
S_base=3/2*U_base*I_base;
T_base=S_base*p/wn;
Z_base=U_base/I_base;


%pu values
i_d=I_d/I_base;
i_q=I_q/I_base;
psi_d=Psi_d/Psi_base;
psi_q=Psi_q/Psi_base;
psi_m=Psi_m/Psi_base;
xd=Xd/Z_base;
xq=Xq/Z_base;
rs=Rs/Z_base;

%Erasing measured values out of interest
i_cut=2;
i_d_cut=[-i_cut 1e-6]; %Fit curve to chosen ...
    values of Id [id_min id_max]
i_q_cut=[-1e-6 i_cut];  %Fit curve to chosen ...
    values of Id [iq_min iq_max]

% find the values of id, iq that the curve ...
    should not be optimized for
i_d_delete1=find((i_d>i_d_cut(2)));
i_d_delete2=find((i_d<i_d_cut(1)));
i_d([i_d_delete1; i_d_delete2])=[];
i_q([i_d_delete1; i_d_delete2])=[];
psi_d([i_d_delete1; i_d_delete2])=[];
psi_q([i_d_delete1; i_d_delete2])=[];
i_q_delete1=find((i_q>i_q_cut(2)));
i_q_delete2=find((i_q<i_q_cut(1)));
i_d([i_q_delete1;i_q_delete2])=[];
i_q([i_q_delete1;i_q_delete2])=[];
psi_d([i_q_delete1;i_q_delete2])=[];
psi_q([i_q_delete1;i_q_delete2])=[];

psi_dm=psi_d-psi_m;

%MTPA const xd, xq
%Residual plots
i_sa=0.00001:0.01:i_cut+0.00001;
%Const xd xq, formula from compendium of Roy ...
    Nilsen
i_da=psi_m/(4*(xq-xd))-sqrt((psi_m/(4*(xq-xd)))^2
+i_sa.^2/2);
i_qa=sqrt(i_sa.^2-i_da.^2);

%CUT THE PLOTS
i_d_fit=[-2 0];
i_q_fit=[0 2];

%Plotting with values up to psi_values for ...
    the flux linkages in the plots
psi_values=0.025;

%Gridding the measured flux linkages to ...
    measured currents
[i_dGrid,i_qGrid] = ...
    meshgrid(linspace(i_d_fit(1),i_d_fit(2),40),
    linspace(i_q_fit(1),i_q_fit(2),40));

% Interpolation Psi_d
psi_dGrid = ...
    griddata(i_d(:),i_q(:),psi_d(:),i_dGrid(:),
i_qGrid(:),'cubic');
psi_dGrid = reshape(psi_dGrid,size(i_dGrid));

% Interpolation Psi_q
psi_qGrid = ...
    griddata(i_d(:),i_q(:),psi_q(:),i_dGrid(:),
i_qGrid(:),'cubic');
psi_qGrid = reshape(psi_qGrid,size(i_dGrid));
```

B. *Plotting FE measured surfaces*

```
main_master
%Plot measured values in dot 3D plot
figure
plot3(i_d,i_q,psi_d,'.')
xlabel('i_d [pu]');
ylabel('i_q [pu]');
zlabel('\psi_d [pu]');
grid on
xlim([i_d_fit(1),i_d_fit(2)]);
ylim([i_q_fit(1),i_q_fit(2)]);


% Fig.1 Surf plot Psi_d
figure
surf(i_dGrid,i_qGrid,psi_dGrid)
xlabel('i_d [pu]');
ylabel('i_q [pu]');
zlabel('\psi_d [pu]');
xlim([i_d_fit(1),i_d_fit(2)]);
ylim([i_q_fit(1),i_q_fit(2)])
colormap(jet)
colorbar

```

```
% Fig.1 Surf plot Psi_q
figure
surf(i_dGrid,i_qGrid,psi_qGrid)
xlabel('i_d [pu]');
ylabel('i_q [pu]');
zlabel('\psi_q [pu]');
xlim([i_d_fit(1),i_d_fit(2)]);
ylim([i_q_fit(1),i_q_fit(2)]);
colormap(jet)
colorbar
```

C. Torque theory plots

```
main_master
load_data=csvread(
'max_torque_theta_one_sweep_220521.csv',5,0);
%1pu
load_data3=csvread(
'max_torque_theta_one_sweep_2pu_250521.csv',5,0);
%2pu
load_data1=csvread(
'max_torque_theta_one_sweep_reluctance_220521.csv'
,5,0);
load_data2=csvread(
'max_torque_theta_one_sweep_PM_220521.csv',5,0);

%Real values
theta=load_data(:,2);
t=load_data(:,10)/T_base;
i_d_max=load_data(:,5)/I_base;
i_q_max=load_data(:,6)/I_base;
%psi_d_max=load_data(:,7)/Psi_base;
%psi_q_max=load_data(:,8)/Psi_base;

%Additional curves
%Magnet disabled, reluctance torque
theta1=load_data1(:,2);
t1=load_data1(:,10)/T_base;

%Rotor iron disabled, PM torque
theta2=load_data2(:,2);
t2=load_data2(:,10)/T_base;

theta3=load_data3(:,2);
t3=load_data3(:,10)/T_base;
i_d_max3=load_data3(:,5)/I_base;
i_q_max3=load_data3(:,6)/I_base;

figure
plot(theta, t,'r','LineWidth',2.0)
hold on
plot(theta1, t1,'r:','LineWidth',2.0)
hold on
plot(theta2, t2,'r--','LineWidth',2.0)
xlim([0, pi])
grid on
xlabel('\theta [rad]')
ylabel('\tau_e [pu]')
legend('\tau_{e, tot}','\tau_{e, ...
    reluctance}','\tau_{e, PM}')
%title('Torque of IPMSM, by reluctance and ...
    PM torque')

%plot(theta,T)
[t_max1, index] = max(t);
theta_max = theta(index);
figure
plot(theta, t,'r', theta_max, t_max1, ...
    'bo','LineWidth',2.0)
hold on
plot(theta,i_d_max,'k:','LineWidth',2.0)
hold on
plot(theta,i_q_max,'k--','LineWidth',2.0)
xlim([0, pi])
grid on
xlabel('\theta [rad]')
ylabel('\tau_e [pu], i [pu]')
legend('\tau_e','\tau_{e, max}','i_d','i_q')
%title('Torque for constant current i_s of ...
    1pu and different load angles')

%plot(theta,T)
[t_max3, index3] = max(t3);
theta_max3 = theta(index3);
figure
plot(theta3, t3,'r', theta_max3, t_max3, ...
    'bo','LineWidth',2.0)
hold on
plot(theta3,i_d_max3,'k:','LineWidth',2.0)
hold on
plot(theta3,i_q_max3,'k--','LineWidth',2.0)
xlim([0, pi])
grid on
xlabel('\theta [rad]')
ylabel('\tau_e [pu], i [pu]')
legend('\tau_e','\tau_{e, max}','i_d','i_q')
%title('Torque for constant current i_s of ...
    2pu and different load angles')
```

D. Cutting values from data set

```
%Erasing measured values out of interest
i_cut=1;
i_d_cut=[-i_cut 0]; %Fit curve to chosen ...
    values of Id [Id_min Id_max]
i_q_cut=[0 i_cut];  %Fit curve to chosen ...
    values of Id [Iq_min Iq_max]
% find the values of Id, Iq that the curve ...
    should not be optimized for
i_d_delete1=find((i_d>i_d_cut(2)));
i_d_delete2=find((i_d<i_d_cut(1)));
i_d([i_d_delete1; i_d_delete2])=[];
i_q([i_d_delete1; i_d_delete2])=[];
psi_d([i_d_delete1; i_d_delete2])=[];
psi_q([i_d_delete1; i_d_delete2])=[];
i_q_delete1=find((i_q>i_q_cut(2)));
i_q_delete2=find((i_q<i_q_cut(1)));
i_d([i_q_delete1;i_q_delete2])=[];
i_q([i_q_delete1;i_q_delete2])=[];
psi_d([i_q_delete1;i_q_delete2])=[];
psi_q([i_q_delete1;i_q_delete2])=[];
psi_dm=psi_d-psi_m;
```

E. Plotting curves of the linear model

```
main_master

psi_d_xGrid=i_dGrid*xd+psi_m; %psi_d ...
    assuming const x_d
psi_q_xGrid=i_qGrid*xq; %psi_q assuming ...
    const x_q

R_dGrid=-psi_dGrid+psi_d_xGrid; %Diff. in ...
    measured and calculated by x_d
R_qGrid=-psi_qGrid+psi_q_xGrid; %Diff. in ...
    measured and calculated by x_q

%%%%%%%%%%%%%%%%%%%%%%
%Plotting for psi_d
```

```

figure
surf(i_dGrid,i_qGrid,psi_dGrid,'FaceColor','r')
hold on
surf(i_dGrid, i_qGrid, ...
    psi_d_xGrid,'FaceColor','b')
hold on
legend({'Measured','Const. x_d'})
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
zlabel('\psi_d [pu]');


figure
pcolor(i_dGrid,i_qGrid,R_dGrid)
hold on
plot(i_da,i_qa,'k','LineWidth',2.0)
hold on
plot(1*exp(1j*(0:pi/200:2*pi)),'k:','LineWidth'
,2.0)
hold on
plot(2*exp(1j*(0:pi/200:2*pi)),'k:','LineWidth'
,2.0)
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
caxis([-psi_values,psi_values])
zlim([-psi_values,psi_values])
colormap(jet)
colorbar

figure
pcolor(i_dGrid,i_qGrid,psi_dGrid)
hold on
plot(i_da,i_qa,'k','LineWidth',2.0)
hold on
plot(1*exp(1j*(0:pi/200:2*pi)),'k:','LineWidth'
,2.0)
hold on
plot(2*exp(1j*(0:pi/200:2*pi)),'k:','LineWidth'
,2.0)
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
colormap(jet)
colorbar

%%%%%%%%%%%%%%%
%Psi_q

% Fig. Surf plot Psi_q for const xq functions
figure
surf(i_dGrid,i_qGrid,psi_qGrid,'FaceColor','r')
hold on
surf(i_dGrid, i_qGrid, ...
    psi_q_xGrid,'FaceColor','b')
hold on
legend({'Measured','Const. x_q'})
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
zlabel('\psi_q [pu]');


figure
pcolor(i_dGrid,i_qGrid,R_qGrid)
hold on
plot(i_da,i_qa,'k','LineWidth',2.0)
hold on
plot(1*exp(1j*(0:pi/200:2*pi)),'k:','LineWidth'
,2.0) %plotting circles 1pu
hold on
plot(2*exp(1j*(0:pi/200:2*pi)),'k:','LineWidth'
,2.0) %plotting circles 2pu
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
caxis([-psi_values,psi_values])
zlim([-psi_values,psi_values])
colormap(jet)
colorbar

figure
pcolor(i_dGrid,i_qGrid,psi_qGrid)
hold on
plot(i_da,i_qa,'k','LineWidth',2.0)
hold on
plot(1*exp(1j*(0:pi/200:2*pi)),'k:','LineWidth'
,2.0)
hold on
plot(2*exp(1j*(0:pi/200:2*pi)),'k:','LineWidth'
,2.0)
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
colormap(jet)
colorbar
```

F. Curve fitting of the simple polynomial model

```
%1. Standard
main_master %Never comment out

%2 The curve fitting is only done within an ...
    area defined in ICutting_plot
%ICutting_plot %If not commented out, the ...
    curve fit will be done within
%the measured area defined in ICutting_plot
%2 end

% % % %3 Curve fitting is only done for a ...
    few mwasurement points in load_datat
% i_d=load_datat(:,5)/I_base;
% i_q=load_datat(:,6)/I_base;
% psi_d=load_datat(:,7)/Psi_base;
% psi_q=load_datat(:,8)/Psi_base;
% psi_dm=psi_d-psi_m;
% % % %3 end

% Building matrix for all I_d I_q values
x1=i_d;

y1=i_q;
y2=i_q.^2;

one= zeros(length(i_d),1)+1;
null=zeros(length(i_d),1);

% %simple fit, reciprocity condition not ...
    taken into account, USED IN PROJECT
Y1=[psi_dm; psi_q];
X1=[x1 null null; null y1 y2];

[B1, BINT, R]=regress(Y1,X1);

d10=B1(1);
q01=B1(2);
q02=B1(3);


R_dm=R(1:length(R)/2, 1);
R_q=R(1+(length(R))/2:length(R), 1);
sumR_dm=sum(abs(R_dm));
sumR_q=sum(abs(R_q));
```

```

%Plotting the curves
psi_d_regGrid=d10*i_dGrid+psi_m;
psi_q_regGrid=q01*i_qGrid+q02*i_qGrid.^2;

%Same plotting as for linear magnetics
```

G. Curve fitting of the best polynomial model

```
%1. Standard
main_master %Never comment out

%2 The curve fitting is only done within an area defined in ICutting_plot
%ICutting_plot %If not commented out, the curve fit will be done within
%the measured area defined in ICutting_plot
%2 end

% % % %3 Curve fitting is only done for a few mwasurement points in load_datat
% i_d=load_datat(:,5)/I_base;
% i_q=load_datat(:,6)/I_base;
% psi_d=load_datat(:,7)/Psi_base;
% psi_q=load_datat(:,8)/Psi_base;
% psi_dm=psi_d-psi_m;
% % % %3 end

% Building matrix for all I_d I_q values
x1=i_d;
x3=i_d.*i_q;
x4=i_q.^2;

y2=i_q;
y4=i_q.^2;
y5=i_d.*i_q.^2;
y6=i_d.^2;

one= zeros(length(i_d),1)+1;
null=zeros(length(i_d),1);

% %Best fit, reciprocity condition not taken into account, USED IN PROJECT
Y1=[psi_dm; psi_q];
X1=[x1 x3 x4 null null null null;null null null y2 y4 y5 y6];

[B1, BINT, R]=regress(Y1,X1);

d10=B1(1);
d11=B1(2);
d02=B1(3);
q01=B1(4);
q02=B1(5);
q12=B1(6);
q20=B1(7);

R_dm=R(1:length(R)/2, 1);
R_q=R(1+(length(R))/2:length(R), 1);
sumR_dm=sum(abs(R_dm));
sumR_q=sum(abs(R_q));

%Plotting the curves
psi_d_regGrid=d10*i_dGrid+
d11*i_dGrid.*i_qGrid+d02*i_qGrid.^2+psi_m;
psi_q_regGrid=q01*i_qGrid+q02*i_qGrid.^2+
q12*i_qGrid.^2.*i_dGrid+ ...
    q20*i_dGrid.^2;
```

H. Curve fitting of the reciprocal polynomial model

```
%1. Standard
main_master %Never comment out

%2 The curve fitting is only done within an area defined in ICutting_plot
%ICutting_plot %If not commented out, the curve fit will be done within
%the measured area defined in ICutting_plot
%2 end

% % %3 Curve fitting is only done for a few mwasurement points in load_datat
% i_d=load_datat(:,5)/I_base;
% i_q=load_datat(:,6)/I_base;
% psi_d=load_datat(:,7)/Psi_base;
% psi_q=load_datat(:,8)/Psi_base;
% psi_dm=psi_d-psi_m;
% % % %3 end

% Building matrix for all I_d I_q values
x1=i_d;
x3=i_d.*i_q;
x4=i_q.^2;
x5=i_q.^3;

y2=i_q;
y3=i_d.*i_q;
y4=i_q.^2;
y5=i_d.*i_q.^2;
y6=i_d.^2;

one= zeros(length(i_d),1)+1;
null=zeros(length(i_d),1);

%RECIPROCITY CONDITION, ALSO INCLUDED IN PROJECT
%Version 3
Y1=[psi_dm; psi_q];
X1=[x1 x3 x4 null null x5*1/3; null 0.5*y6 2*y3 y2 y4 y5];

[B1, BINT, R]=regress(Y1,X1);

d10=B1(1);
d11=B1(2);
d02=B1(3);
q01=B1(4);
q02=B1(5);
q12=B1(6);

R_dm=R(1:length(R)/2, 1);
R_q=R(1+(length(R))/2:length(R), 1);
sumR_dm=sum(abs(R_dm));
sumR_q=sum(abs(R_q));

%Plotting the curves
psi_d_regGrid=d10*i_dGrid+d11*i_dGrid.*i_qGrid
+d02*i_qGrid.^2+1/3*q12*i_qGrid.^3+psi_m;
psi_q_regGrid=q01*i_qGrid+q02*i_qGrid.^2
+q12*i_qGrid.^2.*i_dGrid+1/2*d11*i_dGrid.^2
+2*d02*i_dGrid.*i_qGrid;
```

I. Curve fitting and plotting curves of the exponential model

```
%1. Standard
main_master %Never comment out

% %2 The curve fitting is only done within an area defined in ICutting_plot
```

```
%ICutting_plot %If not commented out, the curve fit will be done within
% %the measured area defined in ICutting_plot, if not used, comment line 42
% %2 end

% % %3 Curve fitting is only done for a few mwasurement points in load_datat
% i_d=load_datat(:,5)/I_base;
% i_q=load_datat(:,6)/I_base;
% psi_d=load_datat(:,7)/Psi_base;
% psi_q=load_datat(:,8)/Psi_base;
% psi_dm=psi_d-psi_m;
% %if not used, comment out line 42, main_master
% % %3 end

%Exponents, chosen values
alpha=2; %7;
beta=2; %4;
Delta=0; %0;
gamma=1; %2;

% Building matrix
x1=(abs(psi_dm).^alpha).*psi_dm;
y1=1/(Delta+2).*abs(psi_dm).^gamma
.*abs(psi_q).^(Delta+2).*psi_dm;
z1=abs(psi_q).^beta.*psi_q;
w1=1/(gamma+2).*abs(psi_dm).^(gamma+2)
.*abs(psi_q).^Delta.*psi_q;
null=zeros(length(i_d),1);

Y1=[i_d; i_q];
X1=[psi_dm null null null y1;
null psi_q null z1 w1]; %a_dd forced to be 0


[B1, BINT, R]=regress(Y1,X1);

a_d0=B1(1);
a_q0=B1(2);
a_dd=B1(3);
a_qq=B1(4);
a_dq=B1(5);

main_master

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plotting the new curves using formula, by id, iq, x, y-axis
i_d_reg=a_d0*psi_dm
+a_dd.*psi_dm.*abs(psi_dm).^alpha+ ...
    a_dq/(Delta+2)*abs(psi_dm).^gamma
    .*abs(psi_q).^(Delta+2).*psi_dm;
i_q_reg=a_q0*psi_q+a_qq.*psi_q.*abs(psi_q).^beta+
    a_dq/(gamma+2)*abs(psi_dm).^(gamma+2)
    .*abs(psi_q).^(Delta).*psi_q;
% Interpolation psi_d
psi_d_regGrid = griddata(i_d_reg(:),i_q_reg(:)
,psi_d(:),i_dGrid(:),...
    i_qGrid(:),'cubic');
psi_d_regGrid = ...
    reshape(psi_d_regGrid,size(i_dGrid));
% % Interpolation psi_q
psi_q_regGrid = griddata(i_d_reg(:),i_q_reg(:)
,psi_q(:),i_dGrid(:),...
    i_qGrid(:),'cubic');
psi_q_regGrid = ...
    reshape(psi_q_regGrid,size(i_qGrid));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fig. Psi_d Plots
figure
surf(i_dGrid,i_qGrid,psi_dGrid,'FaceColor','r')%,
hold on
surf(i_dGrid, i_qGrid, ...
    psi_d_regGrid,'FaceColor','m')%,
legend({'Measured',['Curve fit,' newline ...
    'exponential']})
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
zlabel('\psi_d [pu]');
xlim([-1.95,-0.05])
ylim([0.05,1.95])


figure
pcolor(i_dGrid,i_qGrid,psi_dGrid)
hold on
plot(i_da,i_qa,'k','LineWidth',2.0)
hold on
plot(1*exp(1j*(0:pi/200:2*pi)),'k:'
,'LineWidth',2.0)
hold on
plot(2*exp(1j*(0:pi/200:2*pi)),'k:'
,'LineWidth',2.0)
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
colormap(jet)
colorbar

R_dGrid=-psi_dGrid+psi_d_regGrid;
figure
pcolor(i_dGrid,i_qGrid,R_dGrid)
hold on
plot(i_da,i_qa,'k','LineWidth',2.0)
hold on
plot(1*exp(1j*(0:pi/200:2*pi)),'k:'
,'LineWidth',2.0)
hold on
plot(2*exp(1j*(0:pi/200:2*pi)),'k:'
,'LineWidth',2.0)
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
caxis([-psi_values,psi_values])
zlim([-psi_values,psi_values])
colormap(jet)
colorbar

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Psi_q

figure
surf(i_dGrid,i_qGrid,psi_qGrid,'FaceColor','r')%,
hold on
surf(i_dGrid, i_qGrid, ...
    psi_q_regGrid,'FaceColor','m')%,
legend({'Measured',['Curve fit,' newline ...
    'exponential']})
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
zlabel('\psi_q [pu]');
xlim([-1.95,-0.05])
ylim([0.05,1.95])
zlim([0,1.2])

figure
pcolor(i_dGrid,i_qGrid,psi_qGrid)
hold on
plot(i_da,i_qa,'k','LineWidth',2.0)
hold on
plot(1*exp(1j*(0:pi/200:2*pi)),'k:'
,'LineWidth',2.0)
hold on
```

```
plot(2*exp(1j*(0:pi/200:2*pi)),'k:'
,'LineWidth',2.0)
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
colormap(jet)
colorbar

R_qGrid=-psi_qGrid+psi_q_regGrid;
figure
pcolor(i_dGrid,i_qGrid,R_qGrid)
hold on
plot(i_da,i_qa,'k','LineWidth',2.0)
hold on
plot(1*exp(1j*(0:pi/200:2*pi)),'k:'
,'LineWidth',2.0)
hold on
plot(2*exp(1j*(0:pi/200:2*pi)),'k:'
,'LineWidth',2.0)
hold off
xlabel('i_d [pu]');
ylabel('i_q [pu]');
caxis([-psi_values,psi_values])
zlim([-psi_values,psi_values])
colormap(jet)
colorbar

% %START, find exponents
% %Find which exponents who give the ...
    smallest error
% %Calculate the error
% it=1;
% for i = 0:1:8
%     alpha1=i;
%     for j=0:1:8
%         beta1=j;
%         for k=0:1:8
%             Delta1 =k;
%             for m=0:1:8
%                 gamma1=m;
%
%     x1=(abs(psi_dm).^alpha1).*psi_dm;
%     y1=1/(Delta1+2).*abs(psi_dm).^gamma1
.*abs(psi_q).^(Delta1+2).*psi_dm;
%     z1=abs(psi_q).^beta1.*psi_q;
%     w1=1/(gamma1+2).*abs(psi_dm).^(gamma1+2)
.*abs(psi_q).^Delta1.*psi_q;
%     null=zeros(length(i_d),1);
%
%     Y1=[i_d; i_q];
%     X1=[psi_dm null x1 null y1; null psi_q ...
    null z1 w1];
%
%     [B1, BINT, R]=regress(Y1,X1);
%
%     E(it)=sum(abs(R));
%     para(it,1)=i; para(it,2)=j; ...
    para(it,3)=k; para(it,4)=m;
%             it=it+1;
%             end
%         end
%     end
% end
%
% %Writes which alpha, beta, gamma, Delta ...
    which gives the smallest error
% para(find(E==min(E)),:)
% %END FIND EXPONENTS
```

J. Developing MTPA curves

```
main_master
%FINDING THE COEFFICIENTS by BEST POLYNOMIAL ...
    FUNCTION
% Building matrix for all I_d I_q values
x1=i_d;
x3=i_d.*i_q;
x4=i_q.^2;
y2=i_q;
y4=i_q.^2;
y5=i_d.*i_q.^2;
y6=i_d.^2;
one= zeros(length(i_d),1)+1;
null=zeros(length(i_d),1);
% %Best fit, reciprocity condition not taken ...
    into account, USED IN PROJECT
Y1=[psi_dm; psi_q];
X1=[x1 x3 x4 null null null null;null null ...
    null y2 y4 y5 y6];
[B1, BINT, R]=regress(Y1,X1);
d10=B1(1);d11=B1(2);d02=B1(3);
q01=B1(4);q02=B1(5);q12=B1(6);q20=B1(7);

%FINDING MTPA BY COMSOL MEASUREMENTS
load_data=csvread(
'New_Ld_Lq_max_torque_220521.csv',5,0);
%Real values
theta=load_data(:,2);
t=load_data(:,10)/T_base;
i_d_max=load_data(:,5)/I_base;
i_q_max=load_data(:,6)/I_base;
psi_d_max=load_data(:,7)/Psi_base;
psi_q_max=load_data(:,8)/Psi_base;

theta_sweep=find(theta==pi/2); %Assuming all ...
    sweeps for is have been done
%from pi/2, if not, this value has to be changed

angle_it=-theta_sweep(1)+theta_sweep(2);
it2=length(i_d_max)/angle_it;

for it=1:it2
    t_givenAmp=t(1+angle_it*(it-1):angle_it*it);
    t_max(it,:)=max(t_givenAmp); %Calculated ...
        torque by COMSOL and Arkkios
    t_max_coordi(it,:)= ...
        [i_d_max(angle_it*(it-1)+...
        find(t_givenAmp==max(t_givenAmp))) ...
            i_q_max(angle_it*(it-1)+...
        find(t_givenAmp==max(t_givenAmp)))];
    t_max_coordpsi(it,:)= ...
        [psi_d_max(angle_it*(it-1)+...
        find(t_givenAmp==max(t_givenAmp))) ...
            psi_q_max(angle_it*(it-1)+...
        find(t_givenAmp==max(t_givenAmp)))];
end

i_dit=-1e-6; %initializing the first guess ...
    for the numerical problem
for it=1:length(i_sa) %Implisitt problem
    F=@(i_dit) [-(2 *d10* ...
        i_dit^2)/sqrt(i_sa(it)^2 - i_dit^2) ...
        + ...
        -(psi_m* i_dit)/sqrt(i_sa(it)^2 - ...
            i_dit^2) + ...
        -3* d02* i_dit* sqrt(i_sa(it)^2 - ...
            i_dit^2) + ...
        (d10 *i_sa(it)^2)/sqrt(i_sa(it)^2 - ...
            i_dit^2) + ...
        -3 *d11* i_dit^2 + d11 *i_sa(it)^2 + ...
            4 *q12 *i_dit^3 + ...
        (2 *q01* i_dit^2)/sqrt(i_sa(it)^2 - ...
            i_dit^2) +...
        - (q01* i_sa(it)^2)/sqrt(i_sa(it)^2 ...
            - i_dit^2) +...
```

```
        3* q02 *i_dit^2 - 3 *q20 *i_dit^2 - ...
            2 *q12 *i_dit *i_sa(it)^2+...
        - q02 *i_sa(it)^2];

    i_dit=fsolve(F,i_dit);
    i_dit_vec(it)=i_dit;
end
i_dit=i_dit_vec;
i_qit=sqrt(i_sa.^2-i_dit.^2);

%Plotting the torque for i_s
i_sVec=sqrt(t_max_coordi(:,1).^2
+t_max_coordi(:,2).^2);
%Calculated torque by COMSOL and torque equation
te_c=(t_max_coordpsi(:,1)).*(t_max_coordi(:,2))+
    -(t_max_coordpsi(:,2)).*(t_max_coordi(:,1));
%Calculated torque by analytical formula
te_a=((psi_m-(xq-xd).*i_da).*i_qa); %const ...
    xd xq

te_it=(psi_m.*i_qit+(d10-q01)*i_dit.*i_qit+ ...
    %Best polynomial
    +(d11-q02)*i_dit.*i_qit.^2+d02*i_qit.^3+...
    -q12*i_dit.^2.*i_qit.^2-q20*i_dit.^3);

%Export best id, iq combinations to be run ...
    in COMSOL
csvwrite('myFiled.txt',[i_da*I_base])
csvwrite('myFileq.txt',[i_qa*I_base])
```

K. Plotting MTPA curves

```
IMTPA
%FINDING MTPA BY COMSOL MEASUREMENTS
%Best polynomial function
load_data1=csvread(
'New_Ld_Lq_max_torque_BestComb_010621.csv',5,0);
te_it_com=load_data1(:,10)/T_base;
i_sit_com=sqrt((load_data1(:,5)/I_base).^2
+(load_data1(:,6)/I_base).^2);
%i_qit_com=load_data1(:,6)/I_base;

%Const xd, xq
load_data2=csvread(
'New_Ld_Lq_max_torque_ConstComb_010621.csv',5,0);
te_a_com=load_data2(:,10)/T_base;
i_sa_com=sqrt((load_data2(:,5)/I_base).^2
+(load_data2(:,6)/I_base).^2);
%i_da_com=load_data2(:,5)/I_base;
%i_qa_com=load_data2(:,6)/I_base;

figure
plot(i_sa_com,te_a_com,'b','LineWidth',2.0)
hold on
plot(i_sit_com,te_it_com,'g','LineWidth',2.0)
%hold on
%plot(i_sa,te_a,'b:','LineWidth',2.0)
%hold on
%plot(i_sa,te_it,'g:','LineWidth',2.0)
legend('Arkkios method, const. x_d, x_q',...
    'Arkkios method, best polynomial formula')
xlabel('i_s [pu]');
ylabel('\tau_e [pu]');
grid on
xlim([0,2])


figure
plot(i_sa_com,te_a_com,'b','LineWidth',1.0)
hold on
plot(i_sa,te_a,'b--','LineWidth',1.0)
legend('Arkkios method',...
    'Torque equation')
xlabel('i_s [pu]');
ylabel('\tau_e [pu]');
grid on
xlim([0,2])


figure
plot(i_sit_com,te_it_com,'g','LineWidth',1.0)
hold on
plot(i_sa,te_it,'g--','LineWidth',1.0)
legend('Arkkios method',...
    'Torque equation')
xlabel('i_s [pu]');
ylabel('\tau_e [pu]');
grid on
xlim([0,2])


figure
plot(i_sVec,t_max,'r','LineWidth',1.0)
hold on
plot(i_sVec,te_c,'r--','LineWidth',1.0)
hold on
legend('Arkkios method', ...
    'Torque equation, measured \psi_{dq}, ...
        i_{dq}')
xlabel('i_s [pu]');
ylabel('\tau_e [pu]');
grid on


figure
hold on
scatter(i_d_max,i_q_max,'.')
hold on
scatter(t_max_coordi(:,1),t_max_coordi(:,2),'ro')
grid on
xlabel('i_d [pu]')
ylabel('i_q [pu]')
xlim([-0.05,0]);
ylim([0.05,0.1])




figure
plot(t_max_coordi(:,1),t_max_coordi(:,2),
'r','LineWidth',2.0)
hold on
plot(i_da,i_qa,'b','LineWidth',2.0)
hold on
plot(i_dit,i_qit,'g','LineWidth',2.0)
hold on
plot(1*exp(1j*(0:pi/200:2*pi)),'k--',
'LineWidth',2.0)
hold on
plot(2*exp(1j*(0:pi/200:2*pi)),'k:',
'LineWidth',2.0)
legend('COMSOL','Const. x_d, x_q', 'Best ...
    polynomial', 'i_s=1pu','i_s=2pu')
%title('MTPA trajectories')
xlabel('i_d [pu]');
ylabel('i_q [pu]');
xlim([-2,0]);
ylim([0,2]);
grid on
```

NTNU

Norwegian University of
Science and Technology