



TALLINNA TEHNIKAÜLIKOOL

INSENERITEADUSKOND

Elektroenergeetika ja mehhatroonika instituut

SÜNKROON-RELUKTANTSMOOTORI ROOTORI OPTIMEERIMINE MAGNETAHELA PÕHJAL

SYNCHRONOUS RELUCTANCE MOTOR ROTOR
OPTIMIZATION BASED ON MAGNETIC CIRCUIT

MAGISTRITÖÖ

Üliõpilane: Tõnis Lilleoja

Üliõpilaskood: 176575AAAM

Juhendaja: Ants Kallaste, vanemteadur

Tallinn, 2019

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

“.....” 20.....

Autor:

/ allkiri /

Töö vastab bakalaureusetöö/magistritööle esitatud nõuetele

“.....” 20.....

Juhendaja:

/ allkiri /

Kaitsmisele lubatud

“.....”20.....

Kaitsmiskomisjoni esimees

/ nimi ja allkiri /

LÕPUTÖÖ LÜHIKOKKUVÕTE

Autor: Tõnis Lilleoja

Lõputöö liik: Magistritöö

Töö pealkiri: Sünkroon-reluktantsmootori rootori optimeerimine magnetahela põhjal

Kuupäev: 19.12.2019

100 lk

Ülikool: Tallinna Tehnikaülikool

Teaduskond: Inseneriteaduskond

Instituut: Elektroenergeetika ja mehhatroonika instituut

Töö juhendaja: vanemteadur Ants Kallaste

Töö konsultandid: doktorant-nooremteadur Ekaterina Andriushchenko,
doktorant-nooremteadur Hamidreza Heidari

Sisu kirjeldus:

Antud töö peamiseks eesmärgiks on töötada välja magnetahelal põhinev automaatne optimeerimismeetod sünkroon-reluktantsmootori rootori magnetvoo tõkete sellise asetuse määramiseks, millele vastab maksimaalne d- ja q-telje induktiivsuste suhe. Töö esimeses osas kirjeldatakse sünkroon-reluktantsmootori tööpõhimõtet, pöördemomendi ja võimsusteguri arvutamist ning magnetvootiheduse jaotust õhupilus. Mootori ehitust vaadeldakse osaliselt etteantud 10,3 kW mootori põhjal. Magnetahela koostamise põhimõtet kirjeldatakse eraldi d- ja q-telje korral. Mittelineaarse magnetahela lahendamiseks kasutatakse iteratiivset protsessi ning sõlmpingemeetodit. Staatori hammastele vastava aheldusvoo järgi leitakse eraldi d- ja q-teljele vastavad induktiivsuste väärtused. Induktiivsuste leidmise protsessi automatiseerimiseks on toodud *MATLAB* programmikood lisa 2. Kirjeldatakse rootori magnetvoo tõkete optimeerimisprotsessi, mida vaadeldakse täpsemalt etteantud 10,3 kW mootori põhjal. Optimeerimisprobleem lahendatakse kahel erineval juhul ja lahendamismeetodiks valitakse *genetic algorithm*. Optimeerimisel saadud tulemuste täpsemaks võrdluseks kasutatakse lõplike elementide meetodit. Võrreldes esialgse rootoriga saavutati optimeerimisega esimesel juhul 6,1% kõrgem ja teisel juhul 34,4% kõrgem d- ja q-telje induktiivsuste suhe. Lisaks võrreldakse pulsatsioonimomendi muutust, mis tõusis induktiivsuste suhte suurenedes.

Märksõnad: sünkroon-reluktantsmootor, dq-ristteljestik, magnetahel, sõlmpingemeetod, magnetvoog, induktiivsus, optimeerimine, *genetic algorithm*, pulsatsioonimoment.

ABSTRACT

<i>Author:</i> Tõnis Lilleoja	<i>Type of the work:</i> Master Thesis
<i>Title:</i> Synchronous reluctance motor rotor optimization based on magnetic circuit	
<i>Date:</i> 19.12.2019	100 pages
<i>University:</i> Tallinn University of Technology	
<i>School:</i> School of Engineering	
<i>Department:</i> Department of Electrical Power Engineering and Mechatronics	
<i>Supervisor of the thesis:</i> senior researcher Ants Kallaste	
<i>Consultants:</i> early stage researcher Ekaterina Andriushchenko, early stage researcher Hamidreza Heidari	
<p><i>Abstract:</i></p> <p>The main goal of given work is to design automated optimization method based on magnetic circuit for determining the placement of flux barriers for synchronous reluctance machine in a way to maximize the ratio of d- and q-axis inductances. In the first part of this work synchronous reluctance machine working principle, torque and power factor calculation and magnetic flux density distribution in air gap is described. Motor construction is discussed partly based on given 10,3 kW motor. The principle of magnetic circuit creation is described separately for d- and q-axis. Iterative process is used to find suitable solution for nonlinear magnetic circuit by using node voltage method. By using stator teeth linkage flux the inductances are calculated separately for d- and q-axis. To automate inductance finding process the <i>MATLAB</i> program code is given in appendix 2. Optimization process is described for rotor magnetic flux barriers. Process is stated in more detail based on given 10,3 kW motor. Optimization problem is solved by using <i>genetic algorithm</i> method for two different cases. For more accurate comparison of results the finite element method is used. In comparison with initial rotor the optimized d- and q-axis inductance ratio is 6,1% higher for case 1 and 34,4% higher for case 2. The change of torque ripple is also compared, which increased as inductance ratio got higher.</p>	
<i>Keywords:</i> synchronous reluctance machine, dq-frame, magnetic circuit, node voltage method, magnetic flux, inductance, optimization, <i>genetic algorithm</i> , torque ripple.	

LÕPUTÖÖ ÜLESANNE

Lõputöö teema:	Sünkroon-reluktantsmootori rootori optimeerimine magnetahela põhjal
Lõputöö teema inglise keeles:	Synchronous reluctance motor rotor optimization based on magnetic circuit
Üliõpilane:	Tõnis Lilleoja, 176575
Eriala:	energiamuundus- ja juhtimissüsteemid
Lõputöö liik:	magistritöö
Lõputöö juhendaja:	vanemteadur Ants Kallaste
Lõputöö ülesande kehtivusaeg:	03.01.2020
Lõputöö esitamise tähtaeg:	03.01.2020

Üliõpilane (allkiri)

Juhendaja (allkiri)

Õppekava juht (allkiri)

1. Teema põhjendus

Suure osa kõigist elektrilistest tarbijatest moodustavad elektrimootorid. Sõltuvalt valdkonnast on kasutusel mitmed erinevad tüübid ning suure osa nendest moodustavad asünkroonmootorid. Antud töös vaadeldakse reluktantsmootorit, millega on võimalik saavutada teatud eeliseid klassikalise asünkroonmootori ees. Näiteks kõrgem efektiivsus ja ülekoormatavus, ehituse lihtsus, madalam rootori temperatuur, parem töökindlus. Optimeeritud mootori konstruktsioon võimaldab hoida kokku koostematerjali hulka, parandada võimsustegurit ja tõsta efektiivsust. Kirjanduse allikatest on leitavad mitmed optimeerimise meetodid, aga enamus neist on keskendatud kindlale erijuhule, mis piirab nende mudelite kasutusulatust.

2. Töö eesmärk

Töö esmaseks eesmärgiks on töötada välja optimeerimismeetod reluktantsmootori rootori ehituse lihtsaks ja kiireks esmaseks määramiseks etteantud sisendparameetrite põhjal. Alameesmärgiks on antud optimeerimismeetodi rakendamine etteantud mootorile optimaalse ehitusega rootori leidmiseks.

3. Lahendamisele kuuluvate küsimuste loetelu:

1. Etteantud mootori põhjal staatori ehituse ja mähiste asetuse üleviimine programmikoodi.
2. Etteantud mootori rootori ehituse põhjal koostada programmikood magnetvoo tõkete asetuse määramiseks programmikoodi abil.
3. Mittelineaarse magnetmaterjaliga magnetahela lahendamine.
4. Magnetahelate koostamine ja lahendamine rootori d- ja q-telje asenditele.
5. Magnetahelate lahenduse õigsuse kontroll tarkvara *FEMM* abil.
6. Magnetahelate koostamise laiendamine juhule kui magnetvoo tõkete asetus muudetak.
7. Magnetahelate abil leitud d- ja q-teljele vastavate induktiivsuste suhte ekstreemumpunktide leidmine tarkvara *MathWorks MATLAB* poolt pakutavate optimeerimistööriistadega.

4. Lähteandmed

Füüsilise mootori nimiparameetrid ja joonised selle põhilistest osadest.

Andmed sarnastel teemadel koostatud teadusartiklitest ja uurimustöödest.

Tarkvaras sisalduvad andmed või sellega seotud teabematerjali andmed.

5. Uurimismeetodid

Probleemide lahendamiseks otsitakse infot teemakohastest teadusartiklitest, teadustöödest, raamatutest. Täpsemad ja detailsemad andmed võetakse teemakohastest juhtumisuuringutest, katsetulemustest.

1. Etteantud mootori magnetmudeli koostamiseks kasutatakse programmikoodi, mis kirjutatakse tarkvara *MathWorks MATLAB* abil. *MATLAB* kasutab samanimelist programmeerimiskeelt.
2. Esialgne modelleerimine teostatakse lõplike elementide meetodil. Tarkvara *FEMM* võimaldab suhelda *OctaveFEMM* tööriista abil tarkvaraga *MATLAB*. Programmeerimise keelena on kasutusel *Lua*.
3. Magnetahela koostamine ja lahendamine teostatakse programmikoodi abil tarkvaraga *MATLAB*.
4. Mudeli optimaalse tööpunkti leidmiseks kasutatakse tarkvara *MATLAB*.

6. Graafiline osa

Olulisemad joonised on vajalikud kirjeldamiseks magnetvoo tõkete mõõtmeid ja asetust ning nende muutmist.

Joonised ja skeem on vajalikud magnetahela koostamise ja lahendamise kirjeldamiseks.

7. Töö struktuur

Sisukord

Lühendite ja tähiste loetelu

Sissejuhatus

1. Teoreetiline alus

1.1 Tööpõhimõte

1.2 Mehaaniline tunnusjoon

1.3 *Park'*i ja *Clarke'*i teisendused, pöördemoment

1.4 Mootori aseskeem

1.5 Magnetvootiheduse jaotus õhupilus

2. Reluktantsmootori ehitus

2.1 Staatori ehitus

2.2 Rootori ehitus

3. D-teljele vastav magnetahel

3.1 Ahela koostamine etteantud mootori põhjal

3.2 Takistuste määramine

3.3 Ahela lahendamine

3.4 Ahela koostamine muudetavate sisendparameetritega

4. Q-teljele vastav magnetahel

4.1 Ahela koostamine etteantud mootori põhjal

4.2 Takistuste määramine

4.3 Ahela lahendamine

4.4 Ahela koostamine muudetavate sisendparameetritega

5. Optimeerimismeetodi valik

6. Parandusettepanekud

6.1 Etteantud mootori rootori optimeerimine

6.2 Magnetahela koostamise parandusettepanekud

Kokkuvõte

Kasutatud kirjandus

Lisad

8. Kasutatud kirjanduse allikad

Töö koostamisel kasutatakse infoallikatena raamatuid, teadusartikleid, teadustöid, konverentsimaterjale, tarkavaraga seotud teabematerjale.

Viited peamistele töös kasutatavatele kirjanduse allikatele:

1. Moghaddam, R.R. (2011). Synchronous Reluctance Machine (SynRM) in Variable Speed Drives (VSD) Applications. Doctoral Thesis. Stockholm, Sweden.
2. Hughes, A. (2006). Electric Motors and Drives, Fundamentals, Types and Applications. Third edition. Published by Elsevier Ltd.
3. Bose, B.K. (2002). Modern Power Electronics and AC Drives. The University of Tennessee, Knoxville. Prentice Hall PTR.

9. Lõputöö konsultandid

Vanemteadur Ants Kallaste – lõputöö juhendaja.

Doktorant-nooremteadur Ekaterina Andriushchenko – *MATLAB* optimeerimismeetodi valik.

10. Töö etapid ja ajakava

1. Lähteandmete kogumine, teemaga seotud teooriaga tutvumine, kirjanduse läbitöötamine. (05.2019)
2. Teoreetilise osa kirjutamine. (06.2019)
3. Etteantud mootori magnetmudeli koostamine ja simulatsioon (*FEMM*). (07.2019)
4. Mittelineaarse magnetahela lahendamise programmikoodi koostamine ja katsetamine. (08.2019)
5. Muudetavate magnetvoo tõkete mõõtmete ja asetusega magnetahelate koostamine d- ja q-teljele. (09.2019)
6. Mudelile vastava optimeerimisvariandi valimine ja katsetamine. (10.2019)
7. Mootori laboratoorne parameetrite mõõtmine mudeli tulemuste valideerimiseks. (11.2019)
8. Töö esialgse versiooni esitamine. (25.11.2019)
9. Töö lõpliku versiooni esitamine. (16.12.2019)

SISUKORD

LÕPUTÖÖ LÜHIKOKKUVÕTE.....	3
ABSTRACT.....	4
LÕPUTÖÖ ÜLESANNE	5
LÜHENDITE JA TÄHISTE LOETELU	10
SISSEJUHATUS	12
1. TEOREETILINE ALUS.....	14
1.1 Tööpõhimõte	14
1.2 Mehaaniline tunnusjoon.....	15
1.3 Mootori keskmine pöördemoment ja võimsustegur	16
1.4 Magnetvootiheduse jaotus õhupilus	19
2. SÜNKROON-RELUKTANTSMOOTORI EHITUS	22
2.1 Staatore ehitus.....	23
2.2 Rootore ehitus	24
3. D-TELJE MAGNETAHEL	27
3.1 Ahela koostamine	28
3.2 Takistuste geomeetria määramine	32
4. Q-TELJE MAGNETAHEL	37
4.1 Ahela koostamine	38
4.2 Takistuste geomeetria määramine	41
5. MAGNETAHELA LAHENDAMINE.....	44
5.1 Magnetomotoorjõudude arvutamine.....	44
5.2 Ahela lahendamine sõlmpingemeetodiga	46
5.3 Mittelineaarse ahela lahendi leidmine	48
5.4 Induktiivsuse leidmine	50
6. ROOTORI OPTIMEERIMINE.....	52
6.1 Optimeerimise põhimõte.....	52
6.2 Etteantud 10,3 kW mootore optimeerimine	54
6.3 Induktiivsuste katseline mõõtmine.....	58
6.4 Pulsatsioonimoment	60
KOKKUVÕTE	63
SUMMARY	65
KASUTATUD KIRJANDUS.....	67
Lisa 1 Lihtsustatud voodiagramm ahela sõlmedele vastavate harude määramiseks	71
Lisa 2 <i>MATLAB</i> programmikood induktiivsuste L_d , L_q leidmiseks	74

LÜHENDITE JA TÄHISTE LOETELU

a	magnetilise takistuse laius
A	magnetilise takistuse pindala
B	magnetvootihedus
$\cos \varphi$	võimsustegur
D	diameeter
EL	mootori efektiivne pikkus
f	toitepinge sagedus
F	magnetomotoorjõud
G	magnetiline juhtivus
i	voolu hetkväärtus
i_{avg}	faaside voolude hetkväärtuste aritmeetiline keskmine
i_d	voolu d-komponent
i_k	kogu mõõteahela voolu hetkväärtus
i_q	voolu q-komponent
i_α	voolu α -komponent
i_β	voolu β -komponent
I_m	voolu amplituudväärtus
J	sõlme(magnet)voog
k_w	mähkimistegur
l	magnetilise takistuse pikkus
L	induktiivsus
L_d	induktiivsuse d-komponent
L_k	kogu mõõteahela induktiivsus
L_q	induktiivsuse q-komponent
n_s	sünkroonkiirus
NT	mähise ühe pooli keerdude arv
MMF	magnetomotoorjõud
p	pooluste arv
PH	faaside arv
t	aja hetkväärtus
T_{avg}	keskmine moment
U	magnetiline potentsiaal

U_k	kogu mõõteahela mähise väljaviikudele vastav pinge
R	magnetiline takistus
R_A	faasi A mähise takistus
R_B	faasi B mähise takistus
R_C	faasi C mähise takistus
R_k	kogu mõõteahela takistus
WP	mähiste jaotusviisi tegur
γ	nurk voolu vektori \bar{I} ja d-telje vahel (voolu nurk)
δ_k	koormusnurk
θ	α -telje ja q-telje vaheline pöördenurk
θ_s	sünkroonkiirusele vastav α -telje ja q-telje vaheline pöördenurk
λ	lekkevo koefitsient ($\lambda = 1,12$)
μ_0	vaakumi magnetiline läbitavus ($\mu_0 = 4 \cdot \pi \cdot 10^{-7}$)
μ_r	materjali suhteline magnetiline läbitavus
ν_i	voolu algfaas
τ	mõõtevoolu ajakonstant
φ	nurk voolu vektori \bar{I} ja pinge vektori \bar{U} vahel
Φ	magnetvoog
Ψ	aheldusvoog
ω_s	sünkroonkiirusele vastav voolu nurksagedus

SISSEJUHATUS

Ligikaudu 60...70% kogu tööstuslikust elektrienergiast tarbitakse elektrimootorite poolt [1]. Seega mõjutab nende efektiivsus oluliselt kogu maailma energiatarbimist. Ühtlasi tähendab see potentsiaali säästa ressursse ja tõsta kogu tööstuse efektiivsust. Asünkroonmasin on jätkuvalt kõige laialdasemalt kasutatav mootori tüüp tööstuses (ligikaudu 90%). Selle eelisteks on madal hind, hea kättesaadavus, lihtne konstruktsioon, hea töökindlus, lihtne käitamine. Viimastel aastakümnetel on sünkroon-reluktantsmasin kogunud tähelepanu, sest järjest olulisemaks muutub energia ja ressursside kasutamise efektiivsus [2]. Standardi *IEC 60034-30* kohaselt on sünkroon-reluktantsmasin võimeline saavutama efektiivsuse taseme *IE4* [3]. Seejuures on kõrgem efektiivsus saavutatav sagedusmuunduriga juhtimisel ja suurematel võimsustel (üle 15 kW) [4].

Kirjanduses on laialdaselt võrreldud sünkroon-reluktantsmasinat asünkroonmasinaga (näiteks [5, 1, 6, 4, 7, 8]). Sünkroon-reluktantsmasinal on võrreldes asünkroonmasinaga mõningad eelised. Näiteks on võimalik saavutada kõrgem efektiivsus (puudub rootori vaseskadu), ülekoormatavus, moment ruumala kohta. Lisaks on sünkroon-reluktantsmasinal lihtsam rootori konstruktsioon, mis võimaldab lihtsustada tootmisprotsessi. Suurim puudus on suhteliselt madal võimsustegur, mis tähendab ühtlasi suurema võimsusega sagedusmuundurit [7]. Võimsusteguri tõstmise üks võimalusi on püsomagnetite kasutamine rootoris. Samas tõstab see masina hinda ja muudab projekteerimise keerulisemaks [9].

Sünkroon-reluktantsmasin kuulub tööpõhimõttelt sünkroonmasinate hulka. Staatori ehitus on sama nagu asünkroonmasinal ja sünkroonmasinal. Staatori mähiste toiteks kasutatakse mitmefaasilist vahelduvvoolusüsteemi, mis tagab õhupilus siinuselisele lainekujule lähedase magnetvoo. Kõrgeim efektiivsus saavutatakse mootori elektroonsel juhtimisel sagedusmuunduriga. Rootoris kasutatakse magnetvoo tõkkeid (õhutühimikud), mille järgi on eristatav d-telg (ergutusvoo telg) ja q-telg (pöördemomendi telg). *Park'i* teisenduse abil on leitavad ajas püsivad suurused vastavas dq-teljistikus. See võimaldab lihtsustada arvutusi, probleeme ja mudeleid.

Sünkroon-reluktantsmasina projekteerimise käigus määratakse rootoris magnetvoo tõkete arv ja nende paigutus. Seejuures lähtutakse eelkõige d- ja q-teljele vastavatest induktiivsustest, mis sõltuvad mootori konstruktsioonist. Üheks probleemiks projekteerimisel on nende induktiivsuste optimaalse väärtuse leidmine. Antud töö peamiseks eesmärgiks on töötada välja optimeerimismeetod maksimaalse induktiivsuste erinevuse saavutamiseks. Mida suurem on nende väärtuste erinevus (suhe), seda kõrgem on mootori pöördemoment ja efektiivsus. Seega töös toodud optimeerimismeetod võib olla üheks etapiks mootori projekteerimisel.

Optimeerimisel kasutatakse mootori magnetahelal põhinevat mudelit. Selle abil on võimalik arvutada d- ja q-teljele vastavad magnetvood ja induktiivsused. Töö on ülesehitatud osaliselt etteantud 10,3 kW 4-pooluselise mootori optimeerimiseks. Nimetatud mootori optimeerimisel kasutatakse tarkvara *MathWorks MATLAB* lahendajat *genetic algorithm*. Teisalt on töös toodud optimeerimismeetodit võimalik kohandada ja rakendada kõigile sarnastele juhtudele kus rootori magnetvoo tõkked on kirjeldatavad samade mõõtmete abil. Paljud kirjanduses käsitletud meetodid on keskendatud konkreetsele kitsendatud juhule (näiteks [5, 3, 10, 11, 12, 13, 14]). Töös kasutatava lihtsustatud mudeli eeliseks on selle laiaulatuslik kasutamise võimalus ja lahendamise kiirus. Mootorit vaadeldakse staatilises tühijooksu olekus. Ei arvestata mootori termilisi ja mehaanilisi piiranguid. Kuna antud töös vaadeldakse ainult mootori režiimi, kasutatakse edaspidi mõistet sünkroon-reluktantsmootor.

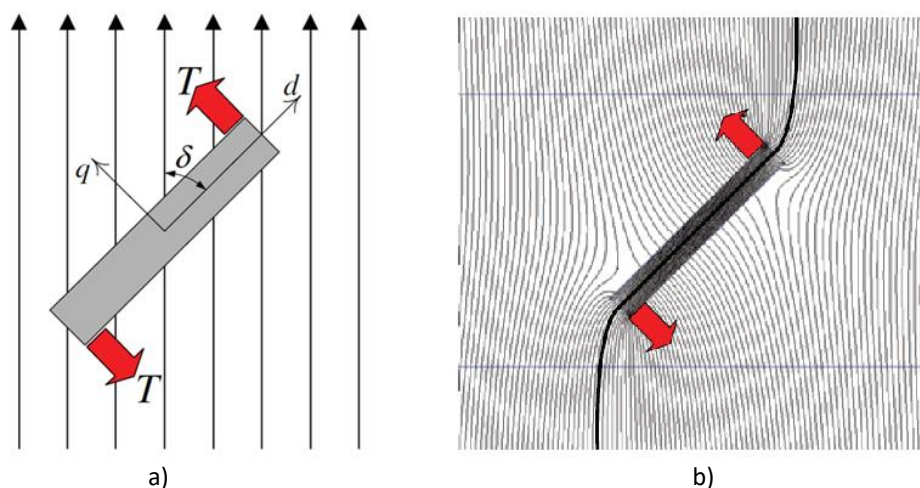
Töö struktuur on järgmine. Peatükis 1 kirjeldatakse lühidalt mootori tööpõhimõtet, momendi ja võimsusteguri avaldumist ning magnetvootiheduse kuju õhupilus. Peatükis 2 kirjeldatakse sünkroon-reluktantsmootori ehitust etteantud 10,3 kW mootori põhjal. Selgitatakse mudelis kasutatavate magnetvoo tõkete mõõtmeid. Peatükis 3 ja 4 kirjeldatakse vastavalt d- ja q-telje magnetahela koostamise põhimõtet. Samuti vaadeldakse ahela takistuste täpsemat määramist. Peatükis 5 kirjeldatakse magnetahela magnetomotoorjõudude arvutamist, sõlmpingemeetodi rakendamist. Ahela sõlmedele vastavate harude määramise protsess on voodiagrammi abil toodud lisa 1. Samuti kirjeldatakse mittelineaarse magnetahela lahendamise protsessi ning viimaks induktiivsuste leidmist. Eeltoodu automatiseerimiseks on toodud *MATLAB* programmikood lisa 2. Peatükis 6 kirjeldatakse rootori optimeerimisprotsessi üldiselt ning etteantud mootori näitel. Tuuakse välja optimeerimise tulemused ja võrdlus tarkvaraga *FEMM* koostatud mudeliga. Lisaks vaadeldakse tulemuste põhjal pulsatsioonimomendi muutust. Mudelite valideerimiseks mõõdetakse etteantud 10,3 kW mootori induktiivsused katseliselt.

1. TEOREETILINE ALUS

1.1 Tööpõhimõte

Sünkroon-reluktantsmootori tööpõhimõte sarnaneb püsिमagnetrootoriga sünkroonmootorile. Mootoril on mähised ainult staatoris. Rotor on koostatud pehmest magnetmaterjalist, milleks üldjuhul on elektrotehnilise terase lehed, mis asetatakse tihedalt kokku mootori võlli suunas. Staator ja selle mähis sarnanevad ansünkroonmootoriga. Staatori mähise ühendamisel 3-faasilisse võrku tekib sünkroonne pöörlev magnetväli. Selle mõjul tekib õhupilus pöörlev magnetväli. Läbi õhupilu liigub staatori poolt tekitatud magnetvoog rootorisse. Staatori ja rootori väljade vahel tekib nihkenurk, mis tekitab rootorile pöördemomendi. Rootori ehitusele vastavalt püüab rootori magnetväli ühilduda staatori magnetväljaga. Staatori poolt tekitatud sünkroonse pöörleva magnetväljaga tagatakse rootori sünkroonkiirusel pöörlemine.

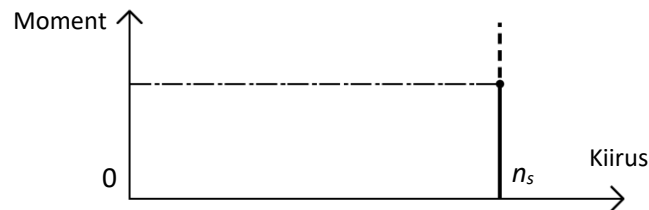
Jõumomendi tekke põhimõte on lühidalt järgmine. Kõrge magnetjuhtivusega materjal asetatakse homogeesse magnetvälja (joonis 1.1, a). Materjali kuju on valitud selliselt, et risti asetsevad mõõtmed (pikkus ja laius) on suuresti erinevad. Materjali suurema pikkuse suunas asetseb d -telg ja lühema suuruse suunas q -telg. Nurk δ magnetvälja jõujoonte ja d -telje vahel on erinev nullist. Materjal moonutab magnetvälja jõujoonte kuju (joonis 1.1, b). Kuna d -telg ei ole magnetvälja jõujoontega samas suunas, mõjub materjalile pöördemoment T sellises suunas, mis püüab minimeerida materjali poolt magnetvoole avaldatavat takistust [7, 15].



Joonis 1.1 Materjalile mõjuvad jõud asetatuna magnetvälja (a) ja magnetvälja moonutus (b) [7]

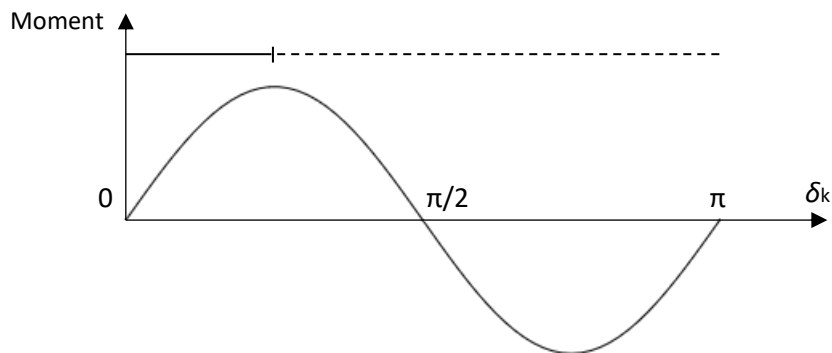
1.2 Mehaaniline tunnusjoon

Sünkroonmootori mehaaniline tunnusjoon ehk kiiruse-pöördemomendi sõltuvus on vertikaalne sirge, mis näitab sünkroonkiirust n_s (joonis 1.2). Rootori koormuse suurendamisel suureneb vastavalt ka rootori tekitatud moment. Koormuse rakendamisel on kindel piir, mille ületamisel rootor väljub sünkroonsest talitlusest ja mootori töörežiim muutub ebastabiilseks (joonisel näidatud kriipsjoonega).



Joonis 1.2 Ideaalse sünkroonmasina mehaaniline tunnusjoon [15]

Eelnimetatud momendi tekkeks on vajalik nihe staatori ja rootori magnetväljade vahel, mis eeldab koormuse rakendamist rootorile. Sellest tulenevalt tekib nende vahel püsiv nihkenurk. Tühijooksu korral on hõõrdejõu ületamiseks vajalik moment ja nihkenurk suhteliselt väikesed. Koormuse suurenedes rootori pöörlemiskiirus aeglustub hetkeks ning nihkenurk suureneb. Seejärel kiirus stabiliseerub. See nihkenurk on tuntud kui koormusnurk δ_k . Joonisel 1.3 on toodud mootori pöördemomendi sõltuvus koormusnurgast δ_k . Stabiilse töö vahemik on $\delta_k = \frac{\pi}{4}$ (näidatud ülal pideva joonega) [15]. Siinkohal vaadeldakse reluktantsmootori elektrilist nurka.



Joonis 1.3 Pöördemomendi sõltuvus koormusnurgast δ_k [16]

1.3 Mootori keskmine pöördemoment ja võimsustegur

Antud töös arvestatakse, et mootori mähiste voolud vastavad 3-faasilisele süsteemile. Vool on siinuselise lainekujuga ja faaside vaheline nihkenurk on 120° . Antud töös arvestatakse ainult voolu põhiharmoonilise komponendi mõju. Ühe faasi kohta on vool kirjeldatav järgnevalt [16]:

$$i_n(t) = I_m \cdot \cos(\omega_s \cdot t + v_i) \quad (1.1)$$

kus i_n – faasi n voolu hetkväärtus, A,
 ω_s – voolu nurksagedus, rad/s,
 I_m – voolu amplituudväärtus, A,
 t – aja hetkväärtus, s,
 v_i – voolu algfaas, rad.

Modelleerimise ja arvutuste teostamisel kasutatakse dq-ristteljestikku. Eriti tähtis on see mootori vektorjuhtimise puhul. Selle eesmärgiks on peamiselt arvutuste ja probleemi lihtsustamine. *Clarke'i* teisendusega viiakse suurused üle liikumatusse ristteljestikku, mille telgedeks on α -telg ja β -telg. *Park'i* teisendusega viiakse suurused $\alpha\beta$ -teljestikust üle pöörlevasse ristteljestikku, mille telgedeks on d-telg ja q-telg. Eeliseks dq-teljestiku korral on püsivad väärtused (ideaalse mootori korral), lihtsam arvutus ja modelleerimine.

Järgnevalt vaadeldakse lühidalt teisenduseks vajalikke arvutusvalemeid. *Clarke'i* teisendusega saab voolude 3-faasilise süsteemi teisendada kahefaasiliseks süsteemiks järgnevate valemitega (analoogne teisendus pingetega) [17]:

$$i_\alpha = \frac{2}{3} \cdot i_a - \frac{1}{3} \cdot i_b + \frac{1}{3} \cdot i_c \quad (1.2)$$

$$i_\beta = \frac{2}{\sqrt{3}} \cdot (i_b - i_c) \quad (1.3)$$

kus i_a, i_b, i_c – vastavad 3-faasilise süsteemi voolude väärtused, A,
 i_α – teisendatud voolu α -komponent, A,
 i_β – teisendatud voolu β -komponent, A.

Park'i teisendusega saab liikumatu kahefaasilise voolude süsteemi teisendada pöörleva ristteljestikuga süsteemiks järgnevate valemitega (analoogne teisendus pingetega):

$$i_d = i_\alpha \cdot \cos \theta + i_\beta \cdot \sin \theta \quad (1.4)$$

$$i_q = -i_\alpha \cdot \sin \theta + i_\beta \cdot \cos \theta \quad (1.5)$$

kus i_d – dq-teljestiku voolu d-komponent, A,

i_q – dq-teljestiku voolu q-komponent, A,
 θ – pöördenurk (nurk α -telje ja q-telje vahel), rad.

Telgede d ja q vaheline nurk on 90° . Pöördemomendile vastab q-telg ja ergutusvoolule d-telg. Selle ristteljestiku eeliseks on ideaalse mudeli korral muutumatud suurused telgede suhtes. Ristteljestik pöörleb üldjuhul sünkroonkiirusel ω_s . Seega sünkroonkiirusele vastav pöördenurk θ_s avaldub [17, 18, 19]:

$$\theta_s = \omega_s \cdot t \quad (1.6)$$

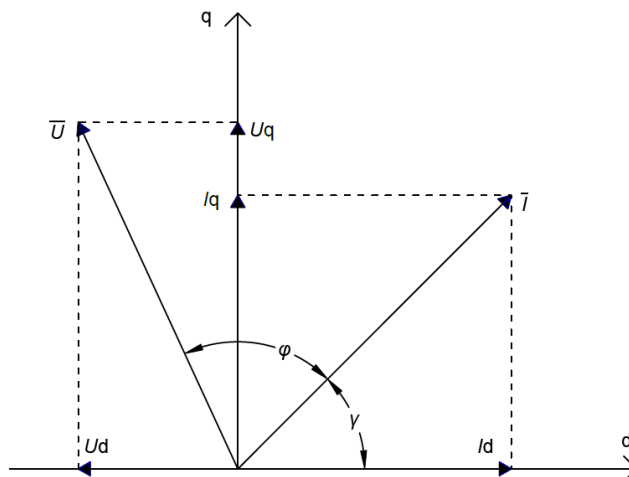
Õhupilu keskmine moment on võimalik välja arvutada järgmiselt [20, 21]:

$$T_{avg} = \frac{3}{2} \cdot \left(\frac{p}{2}\right) \cdot (L_d - L_q) \cdot i_q \cdot i_d \quad (1.7)$$

kus T_{avg} – keskmine moment, Nm,
 p – pooluste arv,
 L_d – mootori induktiivsuse d-komponent, H,
 L_q – mootori induktiivsuse q-komponent, H.

Valemist 1.7 järelduvalt sõltub mootori keskmine moment T_{avg} staatori voolust ja induktiivsuste L_d , L_q väärtustest. Reluktantsmootoril on selgesti eristatavad väljepoolused (vt ptk 2). Sellest järelduvalt on induktiivsuse d- ja q-komponentide erinevus suhteliselt suur (peitpoolustega mootoris $L_d \approx L_q$ [22]). Pöördemoment on seda kõrgem, mida suurem on induktiivsuste erinevus. Sellest järelduvalt on reluktantsmootori väljatöötamisel tähtis saavutada kõrgeim võimalik L_d komponent ja madalaim L_q komponent [16, 20, 23, 21, 7].

Järgnevalt vaadeldakse reluktantsmootori võimsustegurit ($\cos \varphi$). Joonisel 1.4 on toodud mootori vektordiagramm dq-teljestikus [23]. Nurk φ on nurk voolu vektori \vec{I} ja pingve vektori \vec{U} vahel. Nurk γ on nurk voolu vektori \vec{I} ja d-telje vahel (nurk γ on tuntud ka kui voolu nurk).



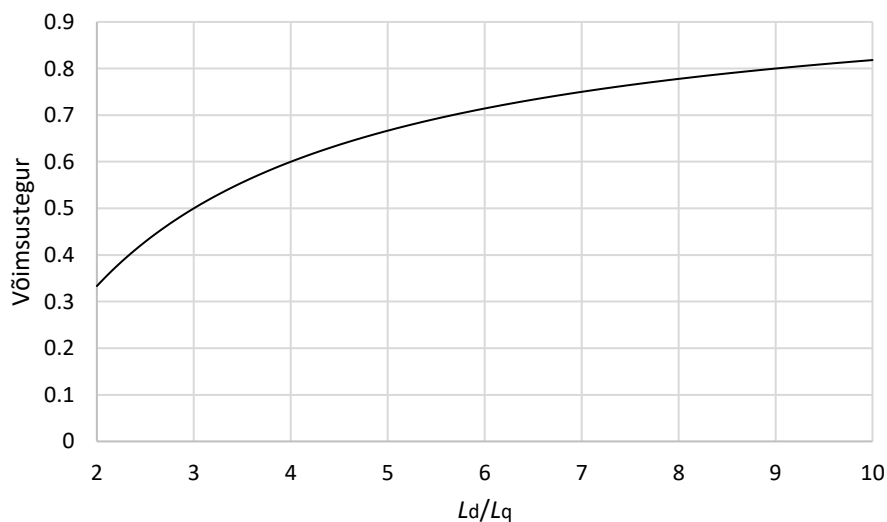
Joonis 1.4 Mootori vektordiagramm

Võimsustegur on reluktantsmootoril üldjuhul suhteliselt madal (võrreldes näiteks asünkroonmootoriga) tulenevalt suuremast reaktiivenergia vajadusest magnetvälja tekitamiseks [24]. Sellest tulenevalt on vajalik kasutada ka suurema võimsusega mootori ajamit. Mootori projekteerimisel on seega tähtis saavutada võimalikult kõrge võimsustegur [23]. Võimsustegurit vaadeldakse antud juhul ainult põhiharmoonilise suhtes. Jättes arvestamata mähise aktiivtakistuse avaldub võimsustegur järgmiselt [23, 21]:

$$\cos \varphi = \left(\frac{L_d}{L_q} - 1 \right) \cdot \sqrt{\frac{\sin 2 \cdot \gamma}{2 \cdot \left(\tan \gamma + \left(\frac{L_d}{L_q} \right)^2 \cdot \cot \gamma \right)}} \quad (1.8)$$

kus φ – voolu ja pinge vektori vaheline nurk, rad,
 γ – voolu vektori ja d-telje vaheline nurk, rad.

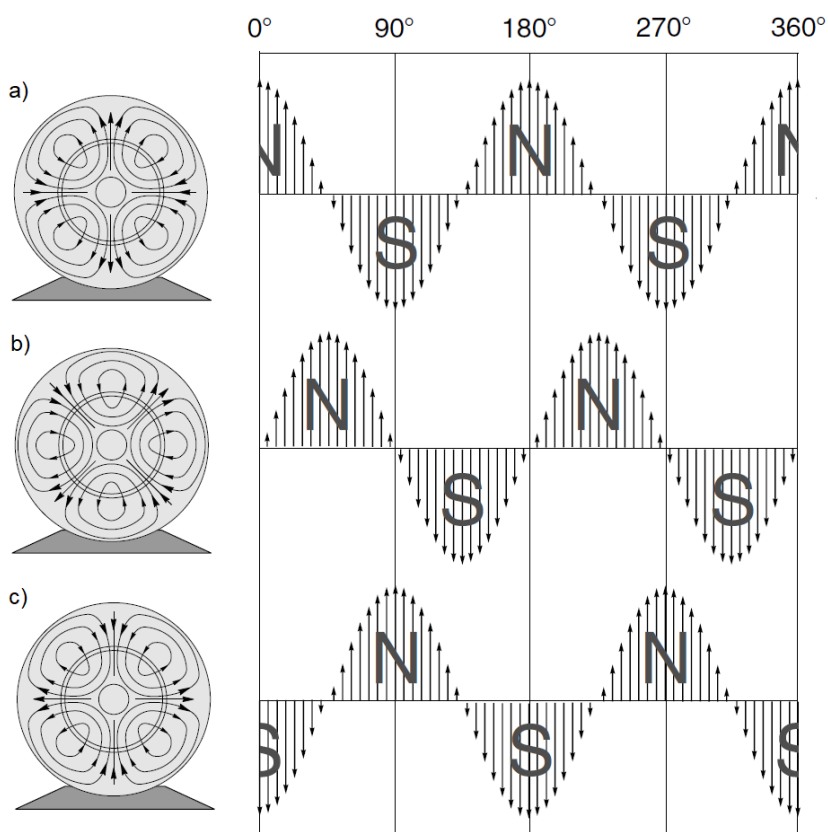
Valemist 1.8 jäeldub, et võimsustegur sõltub induktiivsuse d- ja q-komponentide suhtest L_d/L_q ja voolu nurgast γ . Induktiivsuste L_d , L_q väärtused sõltuvad mootori ehitusest. Voolude i_d , i_q väärtused ja ühtlasi nurk γ sõltuvad mootori juhtsüsteemist [21]. Võimsusteguri suurim väärtus saavutatakse juhul, kui $\gamma = \tan^{-1} \sqrt{L_d/L_q}$ [23, 25]. Sellisel juhul sõltub võimsustegur ainult suhtest L_d/L_q , mis avaldub kui $\cos \varphi = (L_d/L_q - 1)/(L_d/L_q + 1)$. See on graafiliselt toodud joonisel 1.5. Graafikult on näha, et näiteks võimsusteguri 0,75 saavutamiseks on vajalik suhteliselt kõrge suhe $L_d/L_q = 7$. Lisaks on oluline märkida, et mootori keskmine moment T_{avg} sõltub samuti voolu nurgast γ [26].



Joonis 1.5 Võimsusteguri sõltuvus suhtest L_d/L_q

1.4 Magnetvootiheduse jaotus õhupilus

Järgnevalt vaadeldakse lihtsustatud kujul magnetvootiheduse jaotust õhupilus nelja poolusega mootori näitel. Eeldatakse, et mootori toiteks kasutatakse ideaalset kolmefaasilist pinget. Magnetvoo jõujooned, mingil kindlal ajahetkel t , on kujutatud joonisel 1.6, a. Vasakul on toodud illustratiivne vaade jõujoonte asetusest mootoris ning paremal magnetvootiheduse jaotus õhupilus meh. kraadide suhtes. Variant b näitab olukorda, kui jõujoonte süsteem on pöördunud 45° ($t + 5$ ms, kui toitepinge sagedus on 50 Hz). Variant c näitab olukorda, kui jõujoonte süsteem on pöördunud 90° ($t + 10$ ms). Variandi a ja c jõujoonte asetuse on sama, aga poolused on vahetunud. Summaarne magnetvoog õhupilus liigub (pöörleb) sujuvalt ning selle kuju ja suurus on püsivad [27, 15].



Joonis 1.6 Nelja poolusega reluktantsmootori jõujoonte asetuse ja magnetvootiheduse jaotus õhupilus kohal 0° (a), kohal 45° (b) ja kohal 90° (c) [15]

Eelnevalt kirjeldatud nelja poolusega mootori magnetvälja ühele täispöördele vastab toitepinge kaks perioodi. Kui toitepinge sagedus on 50 Hz, siis sünkroonkiirus on vastavalt:

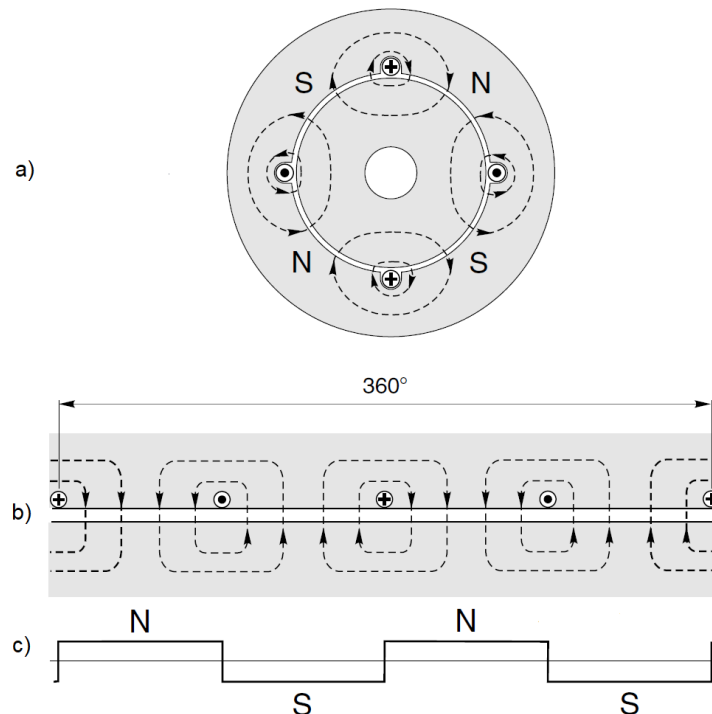
$$n_s = \frac{50}{2} \cdot 60 = 1500 \text{ p/min} \quad (1.9)$$

Seega sünkroonkiirus sõltub pooluste arvust. Avaldisest 1.9 saab tuletada sünkroonkiiruse n_s valemi sõltuvalt pooluste arvust ja sagedusest:

$$n_s = \frac{2}{p} \cdot f \cdot 60 \quad (1.10)$$

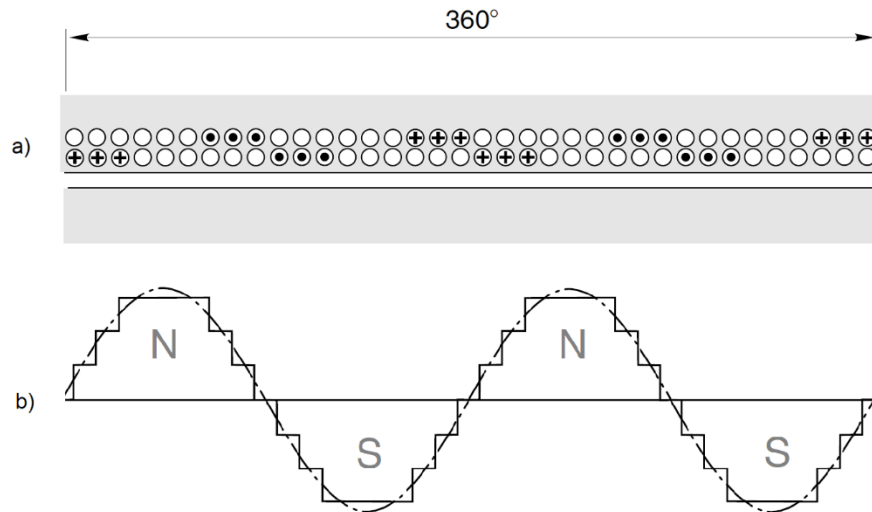
kus f – toitepinge sagedus, Hz.

Staatori mähiste poolt tekitatava magnetvoo täpsemaks vaatluseks on joonisel 1.7, a kujutatud illustratiivne magnetvoo jõujoonte asetus, mis tekib nelja üksiku võrdselt jaotatud juhi poolt. Joonisel 1.7, b on näidatud jõujoonte asetus tasapinnale laotatult. Sellelt nähtub nende võrdne jaotus üksteise suhtes. Jooniselt 1.7, c on näha, et sellise lihtsa jaotusega tekib magnetvootiheduse nelinurkne lainekuju. Mootori sujuvaks tööks on vajalik siinuselise lainekujuga magnetvootihedus.



Joonis 1.7 Nelja juhiga staatori magnetvälja jõujoonte asetus (a), nende tasapinnaline jaotus (b) ning magnetvootiheduse lainekuju (c) (ristiga on tähistatud juhi mineku ja punktiga tuleku suund) [15]

Vootiheduse sujuvamaks muutmiseks kasutatakse mitmekihilist mähist. Kõikidel poolidel on sama mähise samm ja voolu väärtus. Sellisel juhul tekib trepikujuline vootiheduse jaotus, sest juhtide tekitatud vood summeeruvad. See on illustreeritud joonisel 1.8, a, kus on toodud lihtsustatud näide kahekihilisest mähiste paigutusest. Mähise samm on antud juhul 6. Ristide ja punktidega on näidatud ühe faasi mähised. Joonisel 1.8, b on näha resulteeruv vootiheduse lainekuju, mis on lähedasem siinuselisele lainekujule.



Joonis 1.8 Kolmefaasilise staatori ühe faasi juhtide (mähiste) asetuse kahekihilise ehituse korral (a) ning
 resultantne magnetvõtiheduse lainekuju (b) [15]

2. SÜNKROON-RELUKTANTSMOOTORI EHITUS

Antud töös vaadeldav optimeerimismeetod eeldab, et mootori esialgne ehitus ja mõõtmed on eelnevalt määratud. Rootoris peavad olema fikseeritud magnetvoo tõkete arv ning nende esialgne paigutus või paigutuse piirid, mida kirjeldatakse selles peatükis etteantud 10,3 kW mootori põhjal. See informatsioon on vajalik optimeerimismeetodi sisendparameetrite määramiseks. Selles peatükis kirjeldatakse täpsemalt etteantud sünkroon-reluktantsmootori staatori ja rootori ehitust ning mähiste asetust. Etteantud 10,3 kW mootoril on fikseeritud täpsed nimiparameetrid ja mõõtmed. Nimiparameetrid on toodud tabelis 2.1. Staatori ja rootori mõõtmed ning täpsem kirjeldus on toodud vastavalt järgnevates alapeatükkides.

Tabel 2.1 Etteantud mootori nimiparameetrid

Parameeter	Väärtus
Näivvõimsus S , kVA	15,0
Aktiivvõimsus P , kW	10,3
Pinge U , V	400
Vool I , A	22,0
Võimsustegur $\cos\phi$	0,67
Sagedus f , Hz	50
Pöörlemiskiirus n , p/min	1500
Pooluste arv NP	4
Faaside arv PH	3

Töös käsitletavates mudelites vaadeldakse mootorit ideaalsel juhul, jättes välja reaalsest konstruktsioonist tulenevad piirangud ja kõrvalmõjud. Eeldatakse, et materjalid on kõikjal osades homogeenised ning nende mõõtmed täpsed. Staatori mähise osas jäetakse välja mähise otste mõju ja nende põhjustatud lekkevoog. Staatori ehitust ja selle mähise asetust käesolevas töös ei muudeta.

Staatori ja rootori südamik materjalina kasutatakse isoleeritud elektrotehnilise terase M-45 lehti paksusega 0,635 mm ja täiteteguriga 0,98. Teraslehed on laotud nii, et nende tasapind jääb risti rootori teljega. Kogu laotud lehtede paki pikkus on $EL = 156,0$ mm. Staatori ja rootori peamised mõõtmed on kokkuvõtvalt toodud tabelis 2.2.

Tabel 2.2 Staatori ja rootori peamised mõõtmed

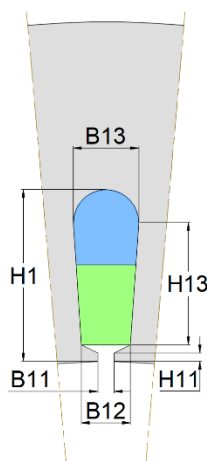
Mõõde	Väärtus
Mootori efektiivne pikkus EL , mm	156,0
Staatori väline diameeter D_{so} , mm	219,0
Staatori sisemine diameeter D_{si} , mm	136,0
Rootori väline diameeter D_{ro} , mm	135,2
Rootori sisemine diameeter D_{ri} , mm	45,0

2.1 Staatori ehitus

Etteantud 10,3 kW mootori staatori peamised mõõtmed on toodud tabelis 2.2. Kasutatakse poolsuletud tüüpi uuret, mille mõõtmed on toodud tabelis 2.3 ja nende asetuse selgitus joonisel 2.1. Üldjuhul moodustab uurde ristlõike pindalast ligikaudu 25...50% isolatsioon. Antud mootori täitetegur on 0,61 (ilma uurde isolatsioonita). Uurete vahele jääva materjali mõõtmetest sõltub staatori hamba laius ja seega ka maksimaalne magnetvoog hambas.

Tabel 2.3 Staatori uurde mõõtmed

Mõõde	Väärtus
$H1$, mm	21,0
$H11$, mm	1,0
$H13$, mm	15,0
$B11$, mm	2,0
$B12$, mm	6,0
$B13$, mm	8,0



Joonis 2.1 Staatori uurde mõõtmete selgitus

Uurete arv staatoris on seotud pooluste arvu, mähkimisviisi ja teiste parameetritega. Antud mootori uurete arv on 36. Kasutatakse kahekihilist mähise asetust. Kahekihilise mähise poolide uurdesse asetamist on kujutatud joonisel 2.1 värvilise pinnaga. Sinine pind kujutab ülemise pooli ristlõiget ja roheline pind alumise pooli ristlõiget. Alade pindalad on seejuures sama suured. Välimise (ülemise) ja sisemise (alumise) kihi mähiste faaside jaotusviis on toodud tabelis 2.4. Siinkohal on toodud 180° staatorist (teine pool on identne). Mähise poolide tähistus on vastavalt faasidele A-X, B-Y, C-Z. Mähise andmed on kokkuvõtvalt toodud tabelis 2.5.

Tabel 2.4 Mähise jaotusviis uures (toodud 180° staatorist)

Kiht	Uurde number																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Väliline	A	A	A	Z	Z	Z	B	B	B	X	X	X	C	C	C	Y	Y	Y
Sisene	A	A	Z	Z	Z	B	B	B	X	X	X	C	C	C	Y	Y	Y	A

Tabel 2.5 Mähise andmed

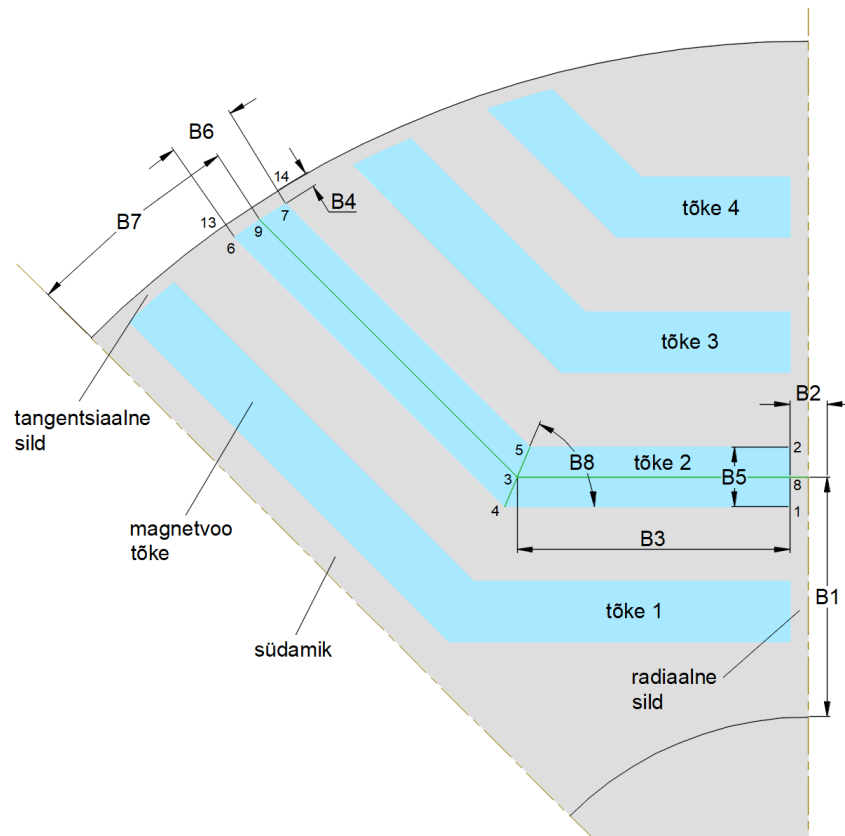
Suurus	Väärtus
Uurete arv <i>NS</i>	36
Kihtide arv	2
Keerdude arv uurdes kihi kohta	23
Keerdude arv uurdes kokku	46
Paralleel traate	5
Traadi läbimõõt (koos isolatsiooniga), mm	0,632
Mähise samm <i>cp</i>	8
Täitetegur (ilma uurde isolatsioonita)	0,61

2.2 Rootori ehitus

Etteantud 10,3 kW mootori rootori peamised mõõtmed on toodud tabelis 2.2. Rootori projekteerimisel on tähtis saavutada d- ja q-teljele vastavate asendite võimalikult kõrge magnetilise takistuse (reluktantsi) erinevus. Selle saavutamiseks kasutatakse vastavaid õhu tühimikke, mis toimivad ühtlasi nii magnetvoo tõkete kui ka voo suunajatena. Mehaaniliste konstruktsiooni piirangute tõttu on tõkete otstesse vajalik jätta piisava laiusega sillad. Antud mootoris kasutatakse ühe pooluse kohta nelja tõket. Tõkete (loendatult alates völlipoolsest tõkkest) mõõtmed on toodud tabelis 2.6 ja nende asetuse selgitus joonisel 2.2 (näidatud pool mootori ühest poolusest). Mõõtmete asetuse valikul on lähtutud võimalusest muuta tõkke kuju piisavalt suures ulatuses. Mõõtmete väärtuste arvutamiseks leitakse punktide koordinaadid, mis on joonisel tähistatud numbritega. Vaadeldava rootori sektori tõkete mõõtmed on identsed teistes sektorites paiknevate tõkete mõõtmetega. Siinkohal on oluline täpsustada, et tõkete kõik punktid peavad jääma sektori ala sisse s.t. ülespoole alumisest sümmeetriateljest ja allapoole ülemisest sümmeetriateljest.

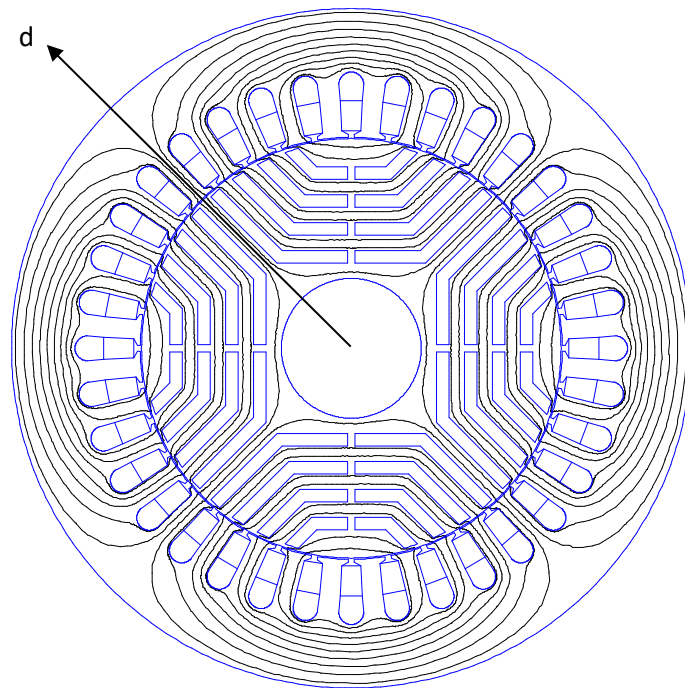
Tabel 2.6 Magnetvoo tõkete mõõtmed

Nr	<i>B1</i> , mm	<i>B2</i> , mm	<i>B3</i> , mm	<i>B4</i> , mm	<i>B5</i> , mm	<i>B6</i> , °	<i>B7</i> , °	<i>B8</i> , °
1	7,0	2,5	21,8860	1,0	4,0	3,4495	3,8759	67,5
2	16,0	2,5	18,1581	1,0	4,0	3,5137	11,6970	67,5
3	25,0	2,5	14,4302	1,0	4,0	3,6535	19,7381	67,5
4	34,0	2,5	10,7023	1,0	4,0	3,8984	28,1921	67,5



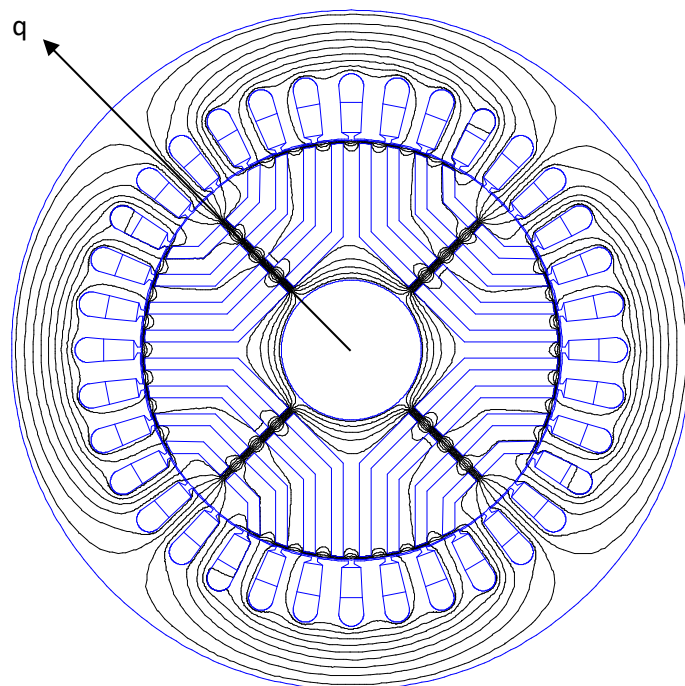
Joonis 2.2 Magnetvoo tõkete mõõtmete selgitus

Magnetvoo tõkete järgi on rootori poolused lihtsasti eristatavad [26]. Vastavalt pooluste asetusele saab paika panna d -telje (ergutusvoo telje) ja q -telje (pöördemomendi telje). Joonisel 2.3 on kujutatud rootor d -teljele vastavas asendis. Sinise kontuuriga on näidatud kõik rootori magnetvoo tõkked ning samuti staatori uurete asetus. Musta kontuuriga on näidatud magnetvoo jooned. Tõkete asetus ja mõõtmed on valitud selliselt, et d -teljele vastav summaarne magnetvoog oleks võimalikult suur. Nähtub, et tõkete kuju järgib magnetvoo joonte asetust juhul, kui rootori tõkked puuduksid (vt joonis 1.6). See on ka üks võimalikest põhimõtetest tõkete optimaalseks paigutuseks [12, 7]. Samuti on tõkete vahelise ala laius püsiv, mis on vajalik ühtlase magnetvootiheduse saavutamiseks sellel alal (terase optimaalne kasutus) [7].



Joonis 2.3 Rootori asetus vastavalt d-teljele

Joonisel 2.4 on kujutatud rootor q-teljele vastavas asendis (staatori osa ei muutu). Sellises asendis on tõkked risti magnetvoo eelistatud liikumistega ning summaarne magnetvoog on minimaalne. Magnetvoo joonte järgi on näha, et suur osa magnetvoost läbib tõkete sildasid, mille magnetiline takistus on väiksem kui magnetvoo tõketel. Tulenevalt terase M-45 mittelineaarsest omadusest tekib sildades magnetiline küllastumine, mis on seda tugevam, mida väiksema laiusga on sillad. Sellest järelduvalt on optimaalse rootori sildade laius minimaalne, mis on piiratud rootori mehaanilise tugevusega.

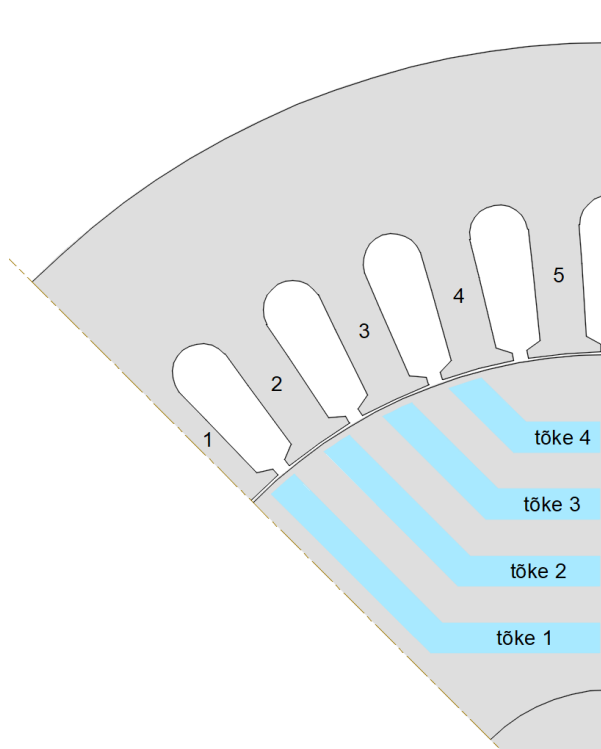


Joonis 2.4 Rootori asetus vastavalt q-teljele

3. D-TELJE MAGNETAHEL

Sünkroon-reluktantsmootori magnetilise osa analüüsimiseks koostatakse lihtsustatud magnetahelal põhinev mudel. Seejuures on mootor staatilises tühijooksu olekus. Selles peatükis vaadeldakse magnetahelat rootori d-teljele vastavas asendis. Seletatakse magnetahela koostamise protsessi ja täpsustatakse magnetiliste takistuste geometria määramist. Ahela koostamisele eelnevalt peab olema fikseeritud magnetvoo tõkete arv ja nende mõõtmed (vt alapeatükk 2.2). Magnetahela koostamine toimub põhimõttel koondada kinnisesse ahelasse magnetomotoorjõu allikad ning magnetmaterjali ja õhu tühimike magnetilised takistused [28]. Ahela struktuur jälgendab magnetvoo asetust mootoris. Tulenevalt ahela struktuurist valitakse lahendamiseks sõlmpingete meetod. Seega ehitatakse ahel üles sõlmede kaupa.

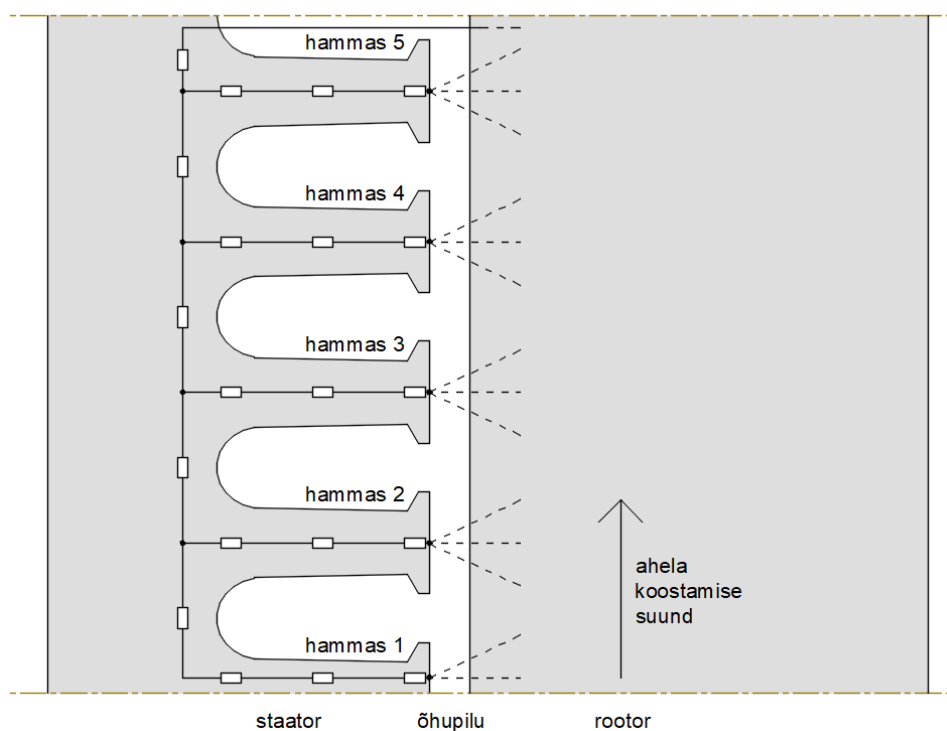
Eelnevast peatükist järeldub, et sünkroon-reluktantsmootor on ristlõikelt (võlli suhtes) sümmeetriline telje suhtes, mis läbib kahte vastastikust poolust. See võimaldab lihtsustada magnetahela koostamist ainult neljandikule kogu mootorist. Samuti on mootor sümmeetriline telje suhtes, mis läbib kahte pooluste vahelist vastastikust punkti. See võimaldab poolitada ahela eelnimetatud telje suhtes, mis tähendab, et vaatluse all on ainult kaheksandik mootorist (üks sektor). Kogu mootorile vastavate suuruste leidmisel korrutatakse lahendamisel saadud tulemused kaheksaga. Parema ettekujutuse mõttes võetakse vaatluse alla mootori ülemine vasakpoolne sektor, mis on toodud etteantud mootori näitel joonisel 3.1. Lisaks on mootor kogu pikkuse ulatuses samasuguse ehitusega. See võimaldab vaadelda probleemi kahemõõtmelises ruumis.



Joonis 3.1 Vaadeldav mootori sektor (etteantud mootori d-telje asend)

3.1 Ahela koostamine

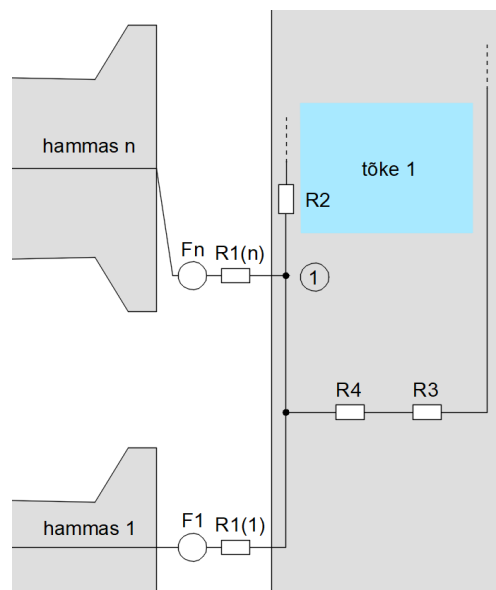
Ahela koostamisel järgitakse, et ahela struktuur oleks võimalikult lihtne ja samas kirjeldaks vaadeldavat olukorda piisavalt täpselt. Magnetmaterjali eraldamisel üksikuteks takistusteks lähtutakse põhimõttest teisendada vastav materjali osa sarnaste mõõtmetega ristkülikuks. Staatoris asuvate takistuste geometria on fikseeritud. Takistuste määramine ja arvutamine on täpsemalt kirjeldatud alapeatükis 3.2. Selles alapeatükis vaadeldakse sõlmega ühenduses olevate ahela harude määramist. Ahela lihtsustamiseks jäetakse välja radiaalsed tökete vahelised sillad, kuna nende mõju on suhteliselt väike (magnetiline potentsiaal radiaalsete sildade läheduses on lähedane nullile). Parema selguse mõttes laotatakse staatori hambad vertikaalselt (joonis 3.2). Rootori magnetahela koostamine toimub vastavalt hammaste tähistuse järjekorrale (näidatud noolega).



Joonis 3.2 D-telje magnetahela koostamise põhimõte

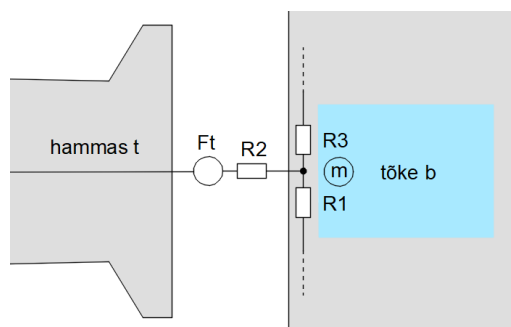
Staatori ikke osas asetsevate sõlmede asukoht on igal juhul fikseeritud. Sõltuvalt ahela struktuurist võib erineda sõlmega ühenduses oleva staatori hambale vastav haru. Kui hamba otsas on sõlm (nagu näidatud joonisel 3.2), siis sisaldab hamba haru ainult hamba takistusi. Kui hamba otsas sõlm puudub (hargnemist ei ole), siis sisaldab hamba haru hamba takistusi, õhupilu takistust ja hambale vastavat magnetomotoorjõudu.

Sõlmega ühenduses olevate harude määramist ja nende järjekorda selgitatakse nelja erineva näitejuhtumi põhjal – esimene sõlm, tõkke välimise otsa kohale jääv sõlm, tõkete vaheline sõlm ja viimane sõlm. Magnetvoo tõkked kujutatakse lihtsuse mõttes ristkülikutena (lühemad küljed vastavad seejuures tõkke otstele). Ahela koostamist alustatakse kõige alumisest sõlmest (sõlm 1), mis on kujutatud joonisel 3.3. Kui tõkke 1 alumine äär asub hamba n kohal, lisanduvad sõlmega harud, milles on vastavalt hammaste järjekorranumbritele magnetomotoorjõud (MMF) $F_{1...n}$ ja õhupilu takistus $R_{1(1...n)}$. Sõlmega on ühendatud tõkke 1 tangentsiaalse silla takistus (või osa sellest) R_2 ja haru, mis sisaldab tõkke 1 alla jääva ala takistusi R_3, R_4 .



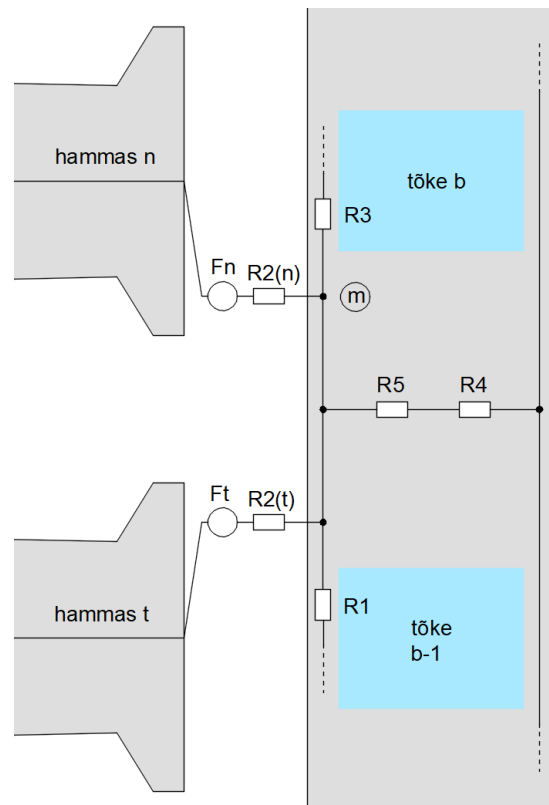
Joonis 3.3 Esimese sõlme ühenduste määramine (d-telg)

Tõkke välimise otsa kohale jääv sõlm (sõlm m) asetseb kahe tangentsiaalse silla takistuse vahel (joonis 3.4). Sõlme allpool harus paikneva tangentsiaalse silla takistuse R_1 geometria on määratud eelneva sõlme alt. Järgmises sõlmega ühenduses olevas harus on vastavalt hamba järjekorranumbrile MMF F_t ja õhupilu takistus R_2 . Sõlme ülevalpool harus paikneb tangentsiaalse silla takistus (või osa sellest) R_3 .



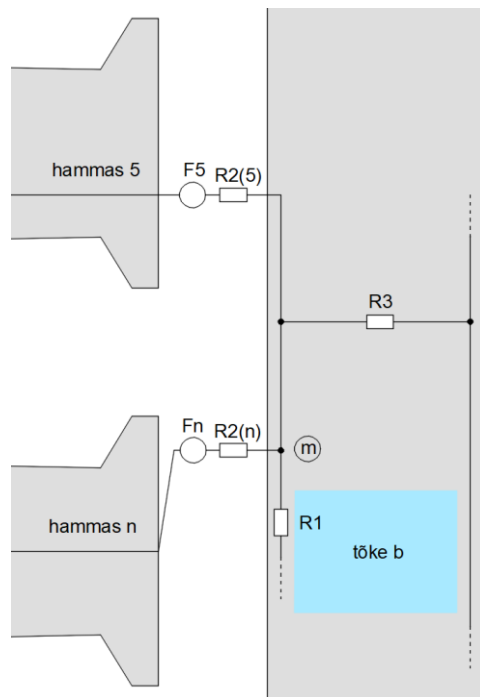
Joonis 3.4 Tõkke välimise otsa kohale jääva sõlme ühenduste määramine (d-telg)

Tõkete vahelise sõlme (sõlm m) ühenduses olevate harude määramise põhimõte on kujutatud joonisel 3.5. Sõlmest allpool paikneva tangentsiaalse silla takistuse R_1 geometria on määratud eelneva sõlme alt. Kui tõkke b alumine äär asub hamba n kohal, lisanduvad sõlmega harud, milles on vastavalt hammaste järjekorranumbritele $MMF F_{t\dots n}$ ja õhupilu takistus $R_{2(t\dots n)}$ (sõltuvalt ka eelmise tõkke ülemise ääre asukohast (tõke $b-1$)). Sõlmest ülevalpool paikneb tangentsiaalse silla takistus (või osa sellest) R_3 . Lisaks on sõlmega ühendatud haru, mis sisaldab tõkete b ja $b-1$ vahele jääva ala takistusi R_4, R_5 .



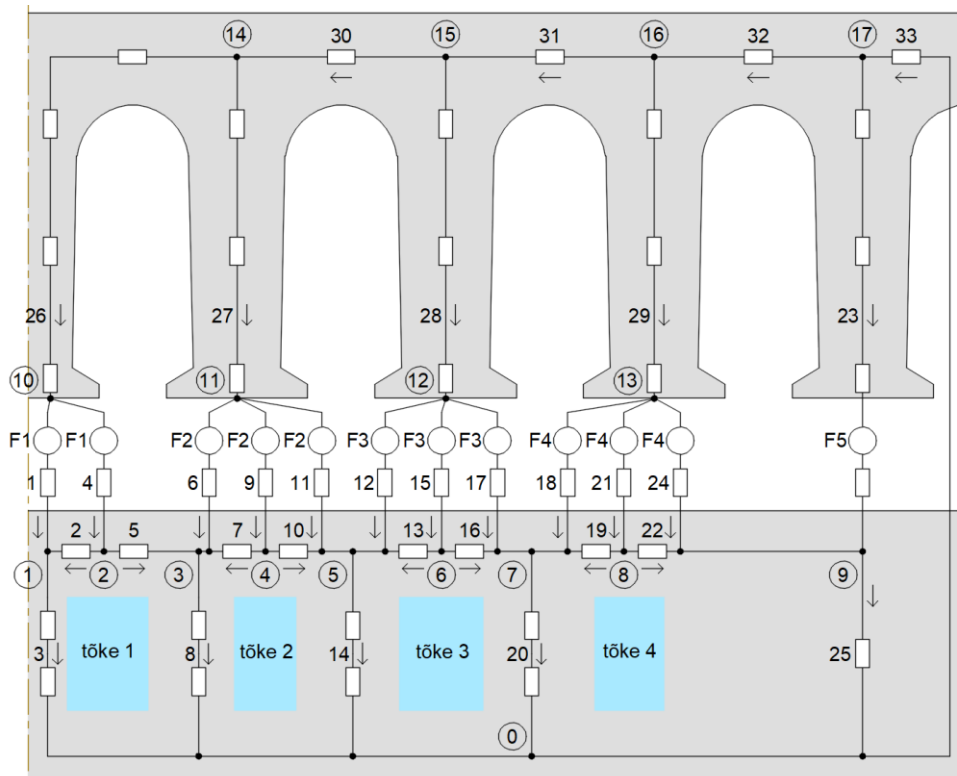
Joonis 3.5 Tõkete vahelise sõlme ühenduste määramine (d-telg)

Ahela viimane sõlm (sõlm m) on kujutatud joonisel 3.6. Sõlmega on ühendatud tangentsiaalse silla takistus R_1 , mille geometria on määratud eelneva sõlme alt. Kui viimase tõkke b ülemine äär asub hamba n kohal, lisanduvad sõlmega harud, milles on vastavalt hammaste järjekorranumbritele $MMF F_{5\dots n}$ ja õhupilu takistus $R_{2(5\dots n)}$. Lisaks on sõlmega ühendatud viimase tõkke b üles jääva ala takistus R_3 .



Joonis 3.6 Viimase sõlme ühenduste määramine (d-telg)

Joonisel 3.7 on toodud d-telje ahela näide etteantud 10,3 kW mootori põhjal. Numbritega, millel on ring ümber on tähistatud sõlmed ja ülejäänud numbritega on tähistatud ahela harud. Nooltega on näidatud voo kokkuleppeline suund harudes. Siinkohal on vajalik täpsustuseks lisada, et ahela lahendamisel on vajalik muuta haru kokkuleppeline voo suund, juhul kui arvutuslik voog selles harus saadakse negatiivne.



Joonis 3.7 D-telje magnetahela näide etteantud 10,3 kW mootori näitel

3.2 Takistuste geometria määramine

Järgnevalt vaadeldakse täpsemalt magnetiliste takistuste arvutamist ja geometria määramist d-telje korral. D-teljele vastava näidisahela takistused on välja toodud joonisel 3.7. Takistuste arvutamisel vaadeldakse eraldi õhu ja terase takistusi, kuna nende magnetiline läbitavus on erinev. Kõige lihtsamal juhul vastab igale takistusele kindla geometriaga risttahukas. Tahuka pikkus on l , mis valitakse magnetvoo liikumise suunas. Risti pikkusega l asetseb pind pindalaga A . Selle pinna laius tähistatakse tähega a ja pinna pikkus on võrdne mootori efektiivse pikkusega EL . Kuna viimane on kõikidel takistustel sama, siis geometria määramisel jäävad otsitavateks suurusteks pikkus l ja laius a . Need määravad ära takistuse pinna samal tasapinnal, mille suhtes mootorit vaadeldakse. Õhu magnetiline takistus mingile kehale arvutatakse järgnevalt [28]:

$$R_{\text{õhk}} = \frac{l}{\mu_0 \cdot EL \cdot a} \quad (3.1)$$

kus $R_{\text{õhk}}$ – õhu magnetiline takistus, $(A \cdot \text{keerdude arv})/\text{Wb}$,

l – magnetilise takistuse pikkus, m,

a – pikkusega l risti asetseva pinna laius, m,

μ_0 – vaakumi magnetiline läbitavus ($\mu_0 = 4 \cdot \pi \cdot 10^{-7}$), H/m.

Terase magnetiline takistus mingile kehale arvutatakse järgnevalt [28]:

$$R_{\text{teras}} = \frac{l}{\mu_0 \cdot \mu_r \cdot EL \cdot a} \quad (3.2)$$

kus R_{teras} – terase magnetiline takistus, $(A \cdot \text{keerdude arv})/\text{Wb}$,

μ_r – materjali suhteline magnetiline läbitavus.

Mudeli täpsuse suurendamiseks arvestatakse tõkete vahelise ala ühe osa pind trapetsina. See võimaldab täpsust tõsta, sest trapetsi korral ei ole magnetvoog sellel alal jaotunud ühtlaselt.

Trapetsi kujulise pinna korral arvutatakse magnetiline takistus terasele järgnevalt [11]:

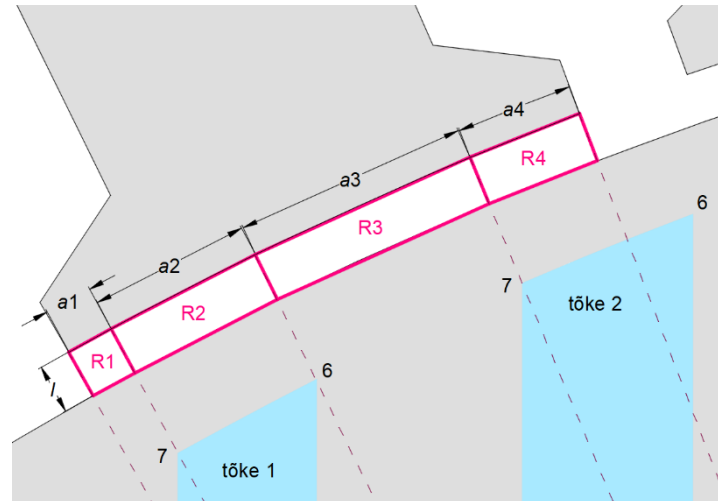
$$R_{\text{teras(trapets)}} = \frac{l}{\mu_0 \cdot \mu_r \cdot EL \cdot (a_{(1)} - a_{(2)})} \cdot \ln\left(\frac{a_{(1)}}{a_{(2)}}\right) \quad (3.3)$$

kus $a_{(1)}, a_{(2)}$ – trapetsi alused.

Täpsustus – antud valem kehtib juhul, kui magnetvoo suund on risti trapetsi alustega.

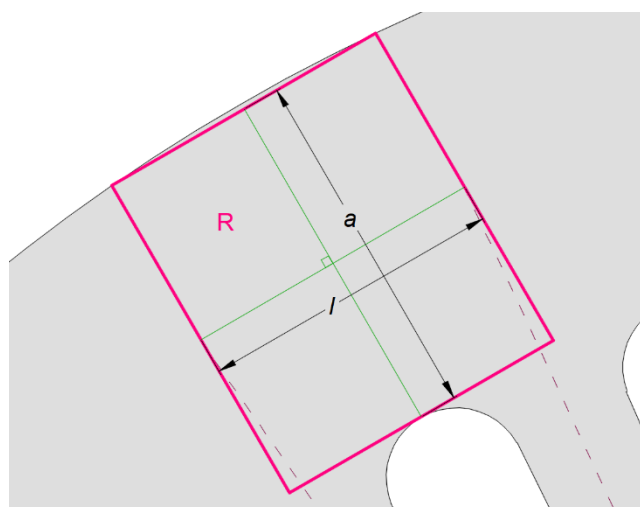
Järgmisena vaadeldakse õhupilu takistuste geometria määramist. Õhupilu takistuste puhul on takistuse pikkus l kõikidel juhtudel võrdne õhupilu pikkusega. Näide nelja õhupilu takistuse R_1, R_2, R_3, R_4 laiuse leidmiseks on toodud joonisel 3.8. Laiused a_1, a_2, a_3, a_4 vastavad staatori sisemise diameetri D_{si} kaarte pikkustele, mis jäävad vastavate nurkade haarade vahele. Nurkade haarad on

näidatud kriipsjoonega ning nende tipud asuvad mootori tsentris. Laiusele a_1 vastava nurga alumine haar on määratud staatori hamba alumise äärega ning ülemine haar tõkke 1 punktiga 7. Laiusele a_2 vastava nurga ülemine haar on määratud tõkke 1 punktiga 6. Laiusele a_3 vastava nurga ülemine haar on määratud tõkke 2 punktiga 7. Laiusele a_4 vastava nurga ülemine haar on määratud staatori hamba ülemise äärega.



Joonis 3.8 Õhupilu takistuste laiuse määramine näite põhjal (d-telg ja q-telg)

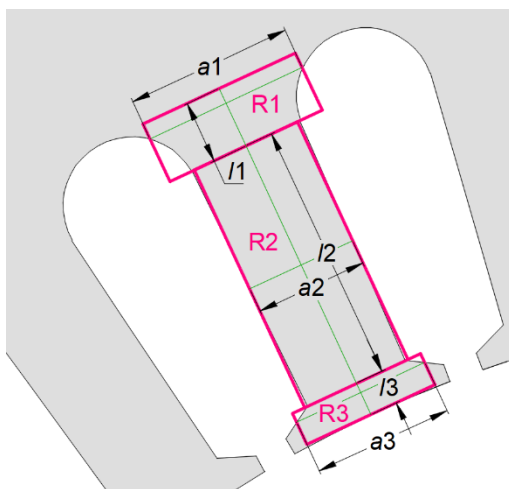
Järgnevalt vaadeldakse staatoris paiknevaid takistusi. Staatori uured ja hambad on identsed, mis lihtsustab takistuste leidmist. Erandiks on hamba 1 takistused (pool laiusest) ja parempoolseim ikke takistus (pool pikkusest), mis on näha joonisel 3.1. Staatori ikke osa geometria määramise selgitus on toodud joonisel 3.9. Kriipsjoonega on näidatud nurga haarad, mis on hammaste sümmeetriatelgedeks ja jagavad ikke võrdseteks osadeks. Nurga tipp on mootori tsentris. Pikkuse l ja laiuse a leidmiseks kasutatakse joonisel rohelisega toodud abilõike.



Joonis 3.9 Staatori ikke osa takistuse määramine

Hamba takistuste R_1 , R_2 , R_3 geometria määramise selgitus on toodud joonisel 3.10. Mõõtmete asetuse selgitamiseks on joonisel rohelisega toodud abilõigud. Pikkus l_1 on määratud uurde kaare

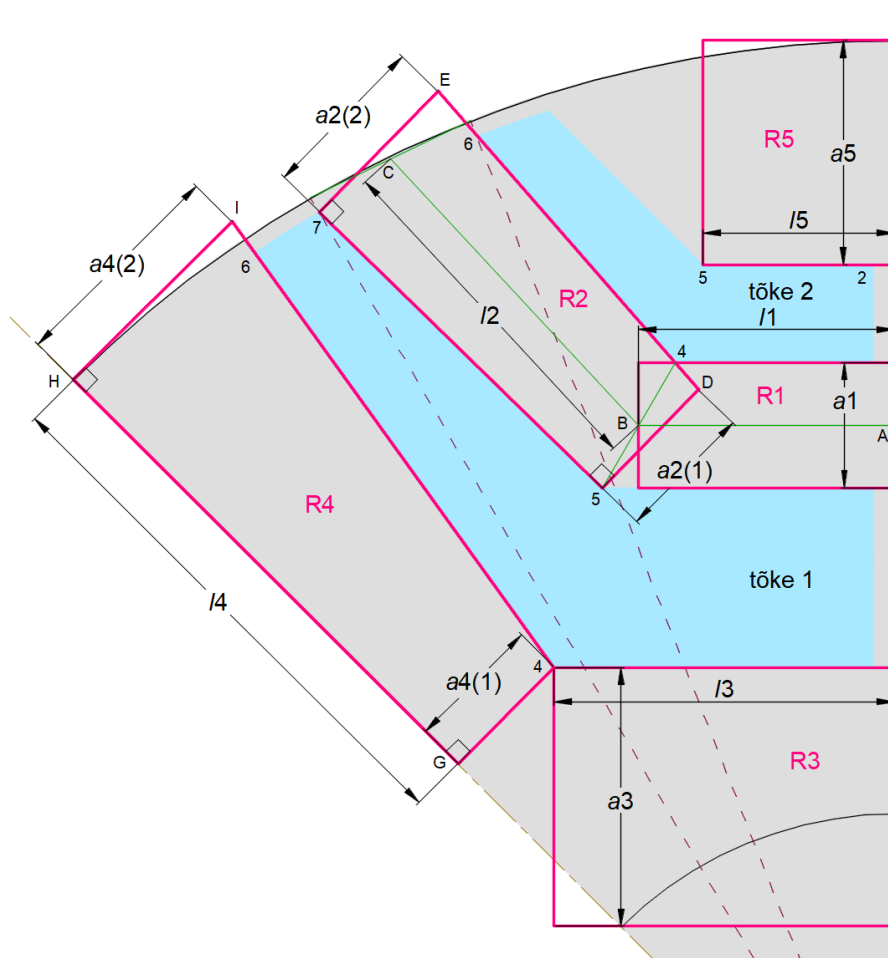
raadiusega. Laius a_1 on määratud poole uurde kaare keskpunktiga. Pikkus l_2 vastab hamba paralleelse osa pikkusele ja laius a_2 selle osa laiuksle. Pikkus l_3 vastab hamba otsa laieneva osa pikkusele ja laius a_3 on määratud hamba trapetsikujulise osa haarde keskpunktidega.



Joonis 3.10 Staatori hamba takistuste määramine

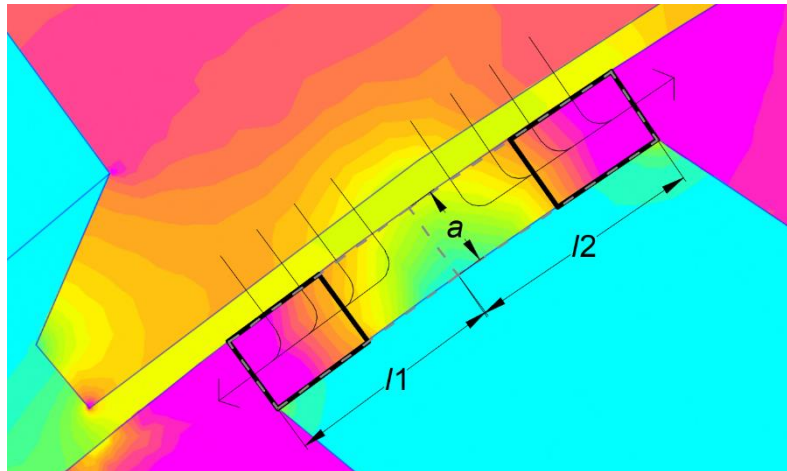
Rootori takistuste määramist vaadeldakse parema selguse mõttes joonisel 3.11 toodud näite põhjal. Kahe tõkke vahel paiknevad takistused on tähistatud R_1 ja R_2 . Nende geometria määratakse tõkke 2 (ülemise tõkke) ja tõkke 1 (alumise tõkke) punktide koordinaatide abil (kõik punktid tähistatud numbritega joonisel 2.2). Geomeetria määramisel kasutatakse abipunkte A, B, C, D, E, mille asetuse selgitamiseks on joonisel rohelisega toodud abilõigud. Punkt C asub abilõigu keskpunktis, mis on määratud kriipsjoonega tähistatud nurga haarde ja rootori välise diameetri D_{ro} lõikepunktidega. Nurga alumine haar on määratud tõkke 1 punktiga 7 ning ülemine haar tõkke 2 punktiga 6 (nurga tipp asub mootori tsentris). Punkt D asub kahe sirge lõikepunktis. Esimene sirge läbib tõkke 1 punkti 5 ja asetseb risti sirgega, mis läbib punkte 5, 7. Teine sirge läbib tõkke 2 punkte 4, 6. Punkt E asub samal sirgel, aga on määratud kolmanda sirgega, mis läbib tõkke 1 punkti 7 ja asetseb risti sirgega, mis läbib punkte 5, 7. Juhul kui tõkkeid on rohkem, määratakse nende vahel paiknevad takistused analoogselt ülalkirjeldatule.

Eraldi vaadeldakse esimese tõkke (joonisel 3.11, tõke 1) all paiknevaid takistusi R_3 ja R_4 . Takistuse R_3 laius a_3 on määratud punktiga 4 ning rootori sisemise diameetri D_{ri} ja alumise sümmeetriatelje lõikepunktiga. Takistuse R_4 geometria määramisel kasutatakse abipunkte G, H, I. Punkt G asub alumisel sümmeetriateljel ja sirgel, mis on risti alumise sümmeetriateljega ja läbib punkti 4. Punkt H asub alumise sümmeetriatelje ja rootori välise diameetri D_{ro} lõikepunktis. Siinkohal on vajalik täpsustuseks lisada, et takistuse R_4 laiused $a_{4(1)}$, $a_{4(2)}$ on pool kogu takistuse ala laiuksel (teine pool jääb mootori alumise sektori sisse). Seda asjaolu on vajalik arvestada magnetahela lahendamisel. Viimase tõkke (joonisel 3.11, tõke 2) üleval paikneva takistuse R_5 geometria määratakse samuti eraldi. Selleks kasutatakse tõkke 2 punkte 2, 5 ja rootori välist diameetrit D_{ro} .



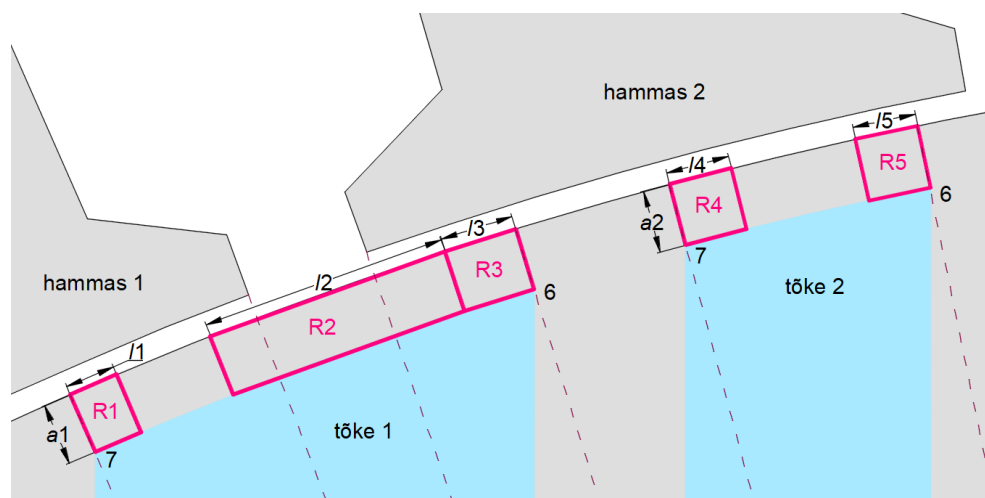
Joonis 3.11 Rootori takistuste määramine näite põhjal (d-telg)

D-teljele vastavate tangentsiaalsete sildade geometria määramise põhimõtet on kirjeldab joonis 3.12, millel on kujutatud olukord kui tõke asub täielikult staatori hamba kohal. Pindalade värvid vastavad pinna osade erinevatele magnetvootiheduse suuruste vahemikele. Sinine värvus vastab vähimale ja roosa värvus suurimale magnetvootihedusele. Värvide abil on selgesti eristatavad kaks silla osa (näidatud kriipsjoonega), milles magnetvoo suund on erinev (tinglik suund näidatud noolega). Mõlema pinna laius a on sama suur. Lihtsuse mõttes võetakse nende pindade pikkused l_1, l_2 võrdseks ($l_1 = l_2$). Kuna magnetvoog siseneb pindadesse kogu pikkuse ulatuses ühtlaselt, siis võetakse arvutuslikuks pikkuseks pool pinna pikkusest ($l_1/2$ ja $l_2/2$). Vastavad pinnad on tähistatud pideva jämejoonega. Järgnevalt täpsustatakse kahte erijuhtu. Esiteks juhul kui tõke jääb täielikult kahe hamba vahele, siis võetakse silla pikkus võrdseks kogu silla pikkusega. Teiseks juhul kui tõke jääb järgmise hamba kohale, siis järgmise hamba kohal võetakse sildade pikkusteks pindade täispikkused (vt joonis 3.13, takistused R_2, R_3).



Joonis 3.12 Tangentsiaalsete sildade geometria määramise põhimõte (d-telg)

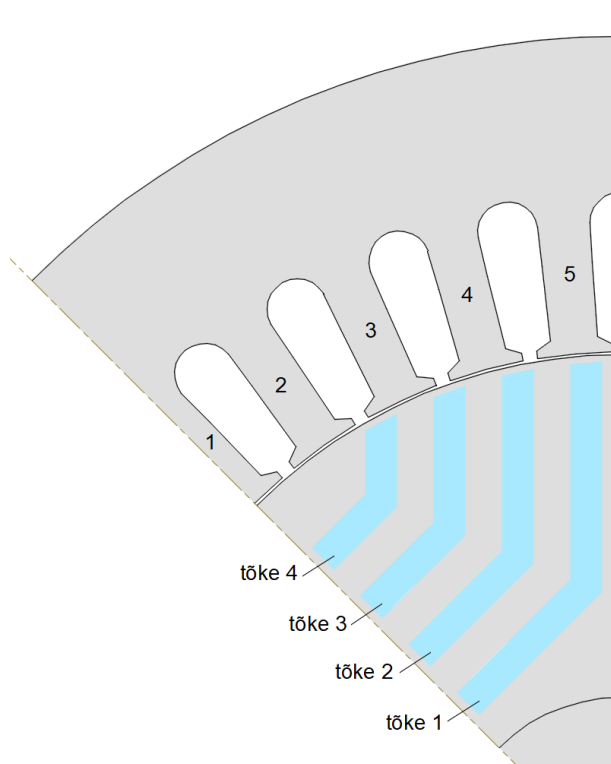
Tangentsiaalsete sildade takistuste täpsemat määramist vaadeldakse joonisel 3.13 toodud näite põhjal. Tõkke 1 kohal paiknevate takistuste R_1, R_2, R_3 laius a_1 vastab tõkke 1 mõõtmele $B4$ (tabel 2.6). Tõkke 2 kohal paiknevate takistuste R_4, R_5 laius a_2 vastab tõkke 2 mõõtmele $B4$. Kõigi takistuste pikkused $l_{1...5}$ sõltuvad rootori välise diameetri D_{r0} kaarte pikkustest, mis on määratud vastavate nurkade haaradega. Nurkade haarad on näidatud kriipsjoonega ning nende tipud asuvad mootori tsentris. Pikkus l_1 on üks neljandik kaarest, millele vastava nurga alumine haar on määratud tõkke 1 punktiga 7 ja ülemine haar staatori hamba 1 ülemise äärega. Pikkuse l_2 esimene osa on samuti neljandik eelnimetatud kaarest. Pikkuse l_2 teine osa on võrdne kaare pikkusega, millele vastava nurga haarad on määratud staatori hammaste vahe suurusega. Pikkuse l_2 kolmas osa on pool kaarest, millele vastava nurga alumine haar on määratud staatori hamba 2 alumise äärega ja ülemine haar tõkke 1 punktiga 6. Pikkus l_3 on pool eelnimetatud kaare pikkusest. Pikkused l_4, l_5 on mõlemad neljandik kaare pikkusest, millele vastava nurga alumine haar on määratud tõkke 2 punktiga 7 ja ülemine haar tõkke 2 punktiga 6.



Joonis 3.13 Tangentsiaalsete sildade takistuste määramine näite põhjal (d-telg)

4. Q-TELJE MAGNETAHEL

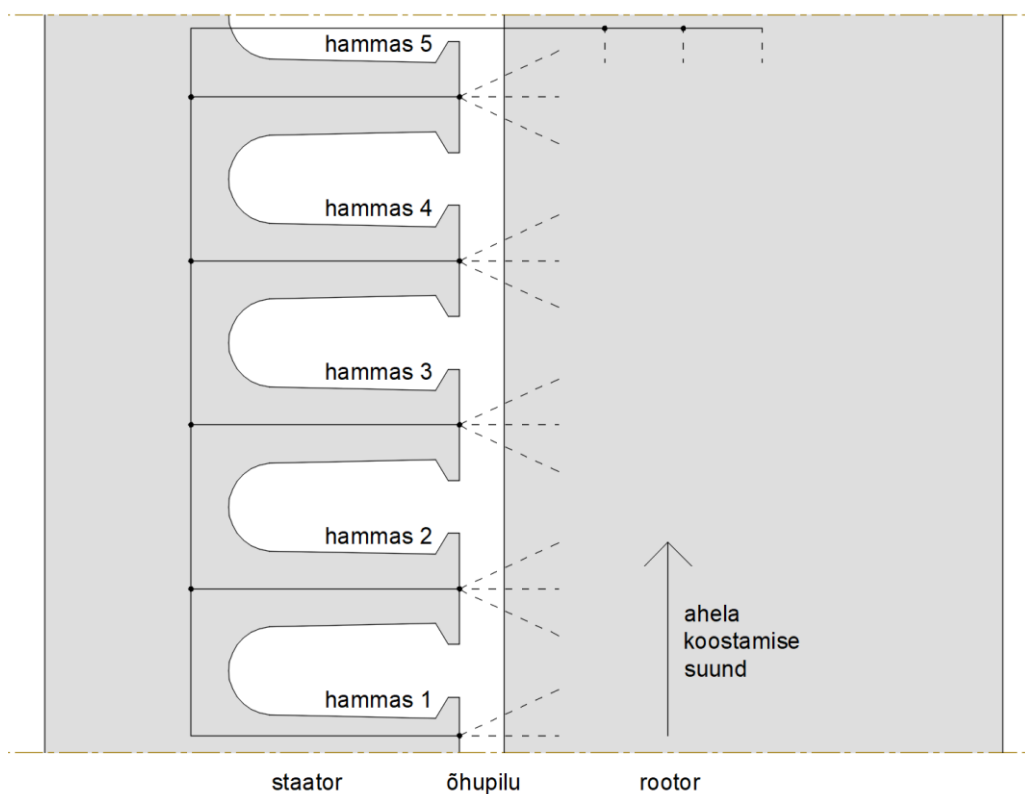
Selles peatükis vaadeldakse magnetahelat rootori q-teljele vastavas asendis. Seletatakse magnetahela koostamise protsessi ja täpsustatakse magnetiliste takistuste geometria määramist. Ahela koostamisele eelnevalt peab olema fikseeritud magnetvoo tõkete arv ja nende mõõtmed (vt alapeatükk 2.2). Magnetahela koostamise peamised põhimõtted jäävad samaks d-teljele vastavale magnetahelale, mis on kirjeldatud eelmises peatükis. Peamisteks erinevusteks on terase takistuste arvestamine ainult sildade ulatuses ja magnetvoo tõkete (õhu) takistuste arvestamine. Sellest tulenevalt on q-telje ahela struktuur mõningal määral erinev. Samamoodi nagu d-telje ahela korral koostatakse ka q-telje magnetahel ainult ühele sektorile kogu mootorist (kaheksandik mootorist). Kogu mootorile vastavate suuruste leidmisel korrutatakse lahendamisel saadud tulemused kaheksaga. Vaatluse alla võetakse mootori ülemine vasakpoolne sektor, mis on toodud etteantud mootori näitel joonisel 4.1.



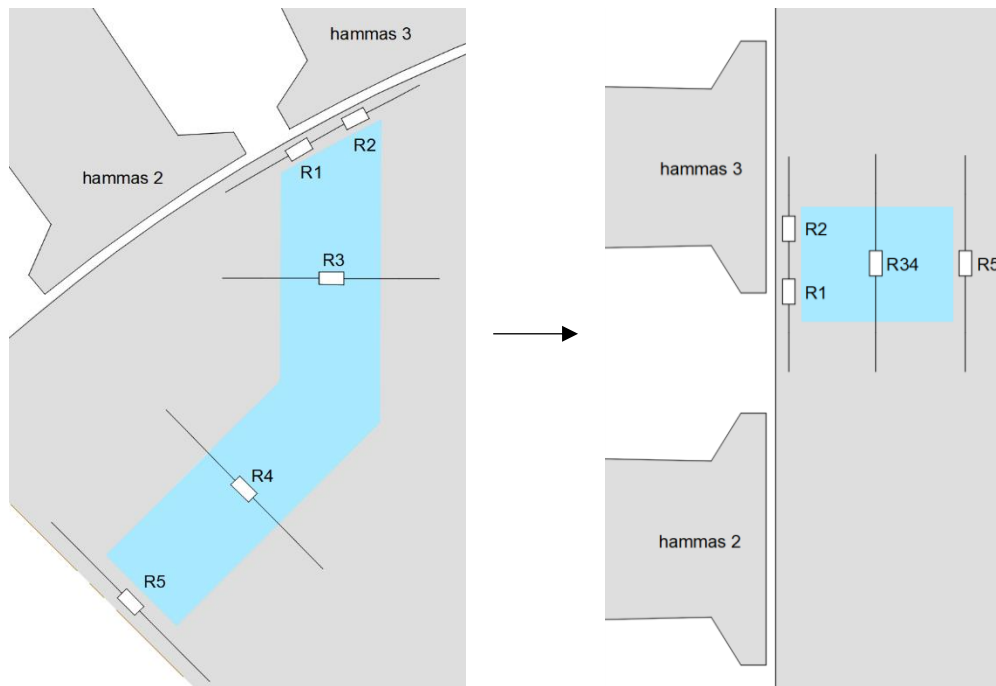
Joonis 4.1 Vaadeldav mootori sektor (etteantud mootori q-telje asend)

4.1 Ahela koostamine

Q-teljele vastava asendi korral on keskmine magnetvootihedus staatori osas ja magnetvoo tõkete vahel üldjuhul mitu korda väiksem kui d-telje korral. Sellest tulenevalt on nende osade terase magnetiline takistus suhteliselt väike. Rootori sildade magnetiline takistus on vastupidiselt palju suurem. See on põhjustatud nende kõrge magnetilise küllastuse tasemest, sest enam magnetvoost läbib rootori sildasid (vt joonis 2.4). Tulenevalt eelnimetatud takistuste suurest erinevusest arvestatakse ahela takistustena ainult õhupilud ning magnetvoo tõkked ja nende sillad. Staatori ja tõkete vaheliste alade takistused võrduvad seega nulliga. Parema selguse mõttes laotatakse staatori hambad vertikaalselt (joonis 4.2). Rootori magnetahela koostamine toimub vastavalt hammaste tähistuse järjekorrale (näidatud noolega). Ahela lihtsustamiseks teisendatakse tõkke kaks paralleelset takistust (näidatud joonisel 4.8) üheks ekvivalentseks takistuseks, mis on illustratiivselt toodud joonisel 4.3.

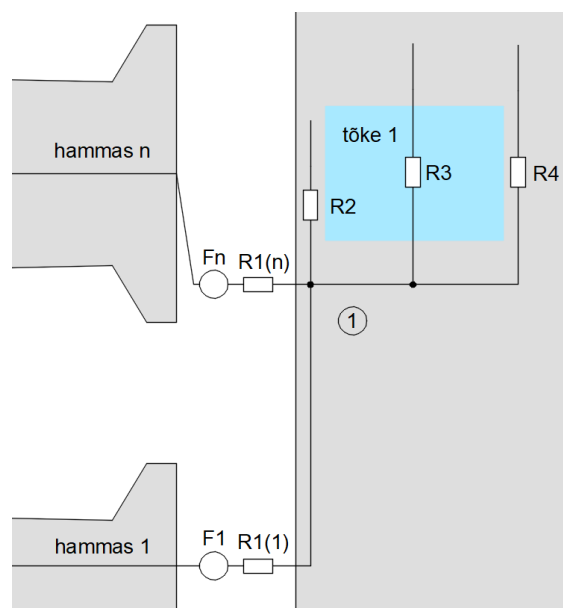


Joonis 4.2 Q-telje magnetahela koostamise põhimõte



Joonis 4.3 Magnetvoo tõkke takistuste R_3 , R_4 teisendus ekvivalentseks takistuseks R_{34}

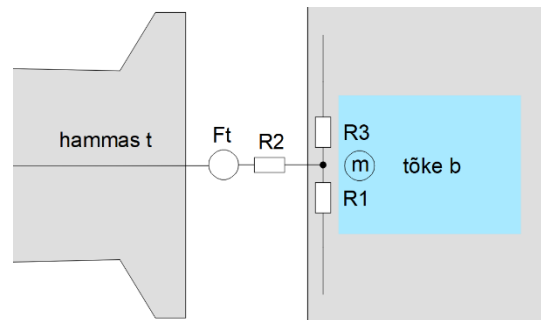
Sõlmega ühenduses olevate harude määramist selgitatakse kolme erineva juhtumi põhjal – esimene sõlm, tõkke välimise otsa kohale jääv sõlm ja tõkete vaheline sõlm. Ahela koostamist alustatakse kõige alumisest sõlmest (sõlm 1), mis on kujutatud joonisel 4.4. Kui tõkke 1 alumine äär asub hamba n kohal, lisanduvad sõlmega harud, milles on vastavalt hammaste järjekorranumbritele $MMF F_{1...n}$ ja õhupilu takistus $R_{1(1...n)}$. Sõlmega on ühendatud tõkke 1 tangentsiaalse silla takistus (või osa sellest) R_2 , tõkke 1 takistus R_3 ja tõkke 1 radiaalse silla takistus R_4 .



Joonis 4.4 Esimese sõlme ühenduste määramine (q-telg)

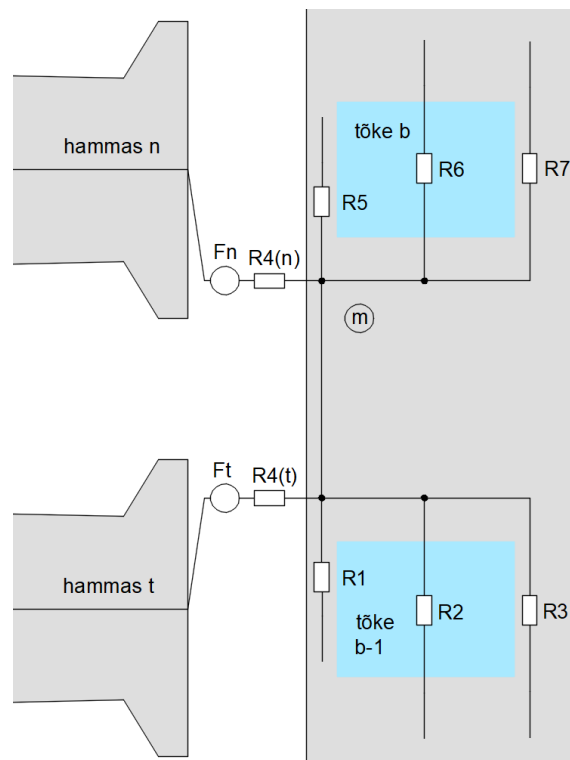
Tõkke välimise otsa kohale jääv sõlm (sõlm m) asetseb kahe tangentsiaalse silla takistuse vahel (joonis 4.5). Sõlme allpool harus paikneva tangentsiaalse silla takistuse R_1 geomeetria on määratud

eelneva sõlme alt. Järgmises sõlmega ühenduses olevas harus on vastavalt hamba järjekorranumbrile $MMF F_t$ ja õhupilu takistus R_2 . Sõlme ülevalpool harus paikneb tangentsiaalse silla takistus (või osa sellest) R_3 .



Joonis 4.5 Tõkke välimise otsa kohale jääva sõlme ühenduste määramine (q-telg)

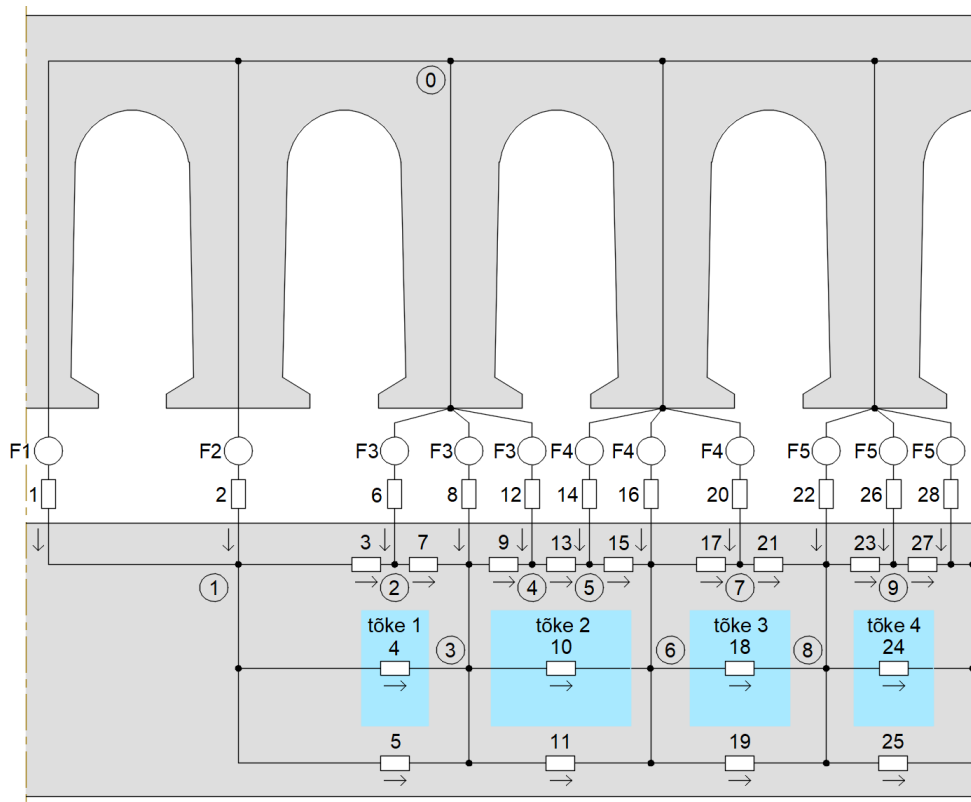
Tõkete vahelise sõlme (sõlm m) ühenduses olevate harude määramise põhimõtte on kujutatud joonisel 4.6. Sõlmest allpool paiknevate takistuste R_1 , R_2 , R_3 geometria on määratud eelneva sõlme (või sõlmede) alt. Kui tõkke b alumine äär asub hamba n kohal, lisanduvad sõlmega harud, milles on vastavalt hammaste järjekorranumbritele $MMF F_{t...n}$ ja õhupilu takistus $R_{4(t...n)}$ (sõltuvalt ka eelmise tõkke ülemise ääre asukohast (tõke $b-1$)). Sõlmest ülevalpool paiknevad tõkkele b vastavad takistused – tangentsiaalse silla takistus R_5 , tõkke takistus R_6 ja radiaalse silla takistus R_7 .



Joonis 4.6 Tõkete vahelise sõlme ühenduste määramine (q-telg)

Joonisel 4.7 on toodud q-telje ahela näide etteantud 10,3 kW mootori põhjal. Numbritega, millel on ring ümber on tähistatud sõlmed ja ülejääänud numbritega on tähistatud ahela harud. Nooltega

on näidatud voo kokkuleppeline suund harudes. Siinkohal on vajalik täpsustuseks lisada, et ahela lahendamisel on vajalik muuta haru kokkuleppeline voo suund, juhul kui arvutuslik voog selles harus saadakse negatiivne.

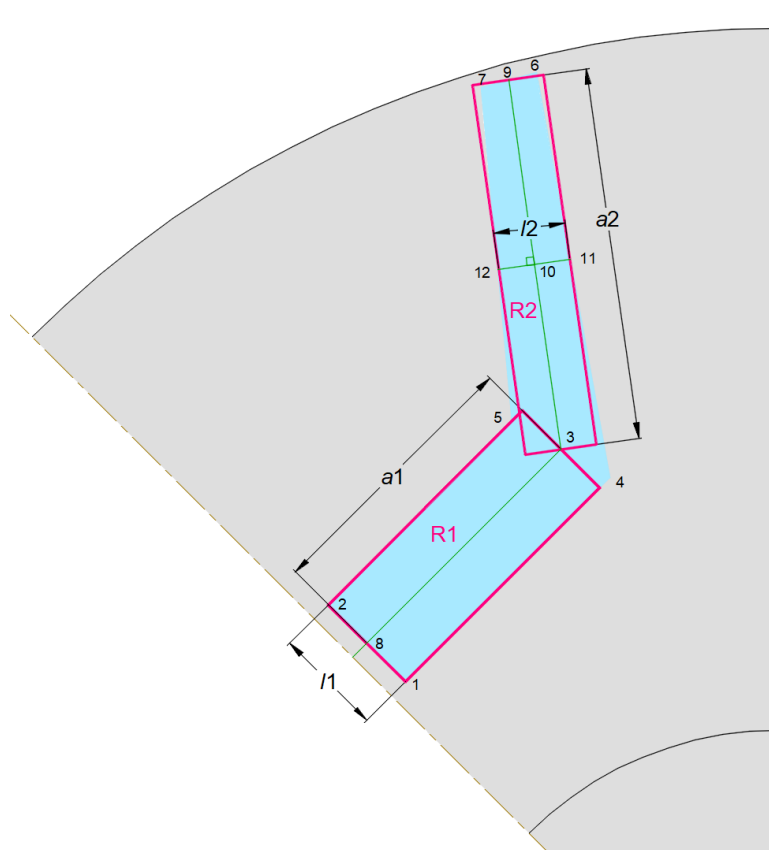


Joonis 4.7 Q-telje magnetahel etteantud 10,3 kW mootori näitel

4.2 Takistuste geomeetria määramine

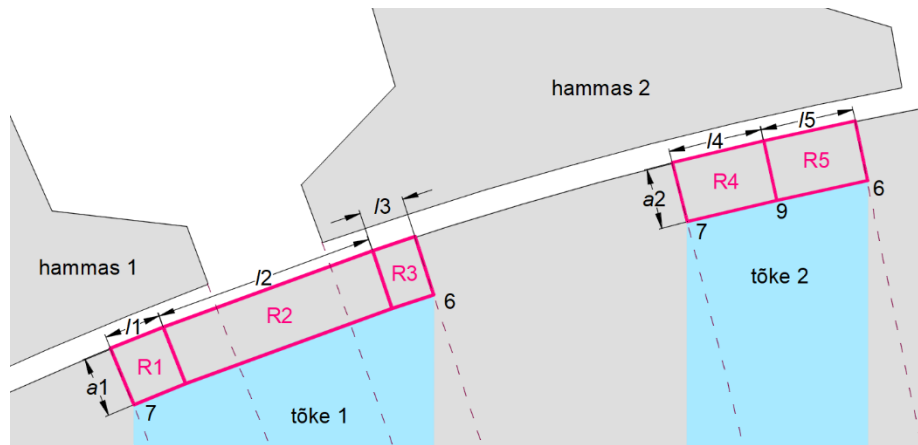
Järgnevalt vaadeldakse täpsemalt magnetiliste takistuste geomeetria määramist q-telje korral. Q-teljele vastava näidisahela takistused on välja toodud joonisel 4.7. Magnetiliste takistuste arvutusvalemid õhu ja terase jaoks on toodud alapeatükis 3.2, mis jäävad samaks q-telje puhul. Seejuures takistuse geomeetria arvutatakse ainult risttahuka põhjal. Takistus sõltub vastava pinna pikkusest l ja laiusest a . Samuti jääb q-telje korral samaks õhupilu takistuste määramise põhimõte (vt alapeatükk 3.2).

Magnetvoo tõkke takistuse määramist vaadeldakse parema selguse mõttes joonisel 4.8 toodud näite põhjal. Tõke jaotatakse kaheks osaks, millele vastavad takistused on tähistatud R_1 ja R_2 . Takistuste geomeetria leidmisel kasutatakse joonisel numbritega tähistatud punktide koordinaate, mis on eelnevalt fikseeritud. Punktide asetuse selgitamiseks on joonisel rohelisega toodud abilõigud. Täpsustus – punkt 10 asub lõigu keskpunktis, mis määratud punktidega 3, 9.



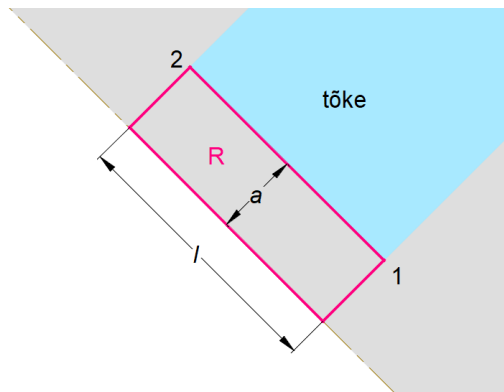
Joonis 4.8 Magnetvoo tõkke takistuste määramine näite põhjal (q-telg)

Tangentsiaalsete sildade takistuste määramist vaadeldakse joonisel 4.9 toodud näite põhjal. Tõkke 1 kohal paiknevate takistuste R_1, R_2, R_3 laius a_1 vastab tõkke 1 mõõtmele B_4 (tabel 2.6). Tõkke 2 kohal paiknevate takistuste R_4, R_5 laius a_2 vastab tõkke 2 mõõtmele B_4 . Kõigi takistuste pikkused $l_{1...5}$ sõltuvad rootori välise diameetri D_{r0} kaarte pikkustest, mis on määratud vastavate nurkade haaradega. Nurkade haarad on näidatud kriipsjoonega ning nende tipud asuvad mootori tsentris. Pikkus l_1 on pool kaare pikkusest, millele vastava nurga alumine haar on määratud tõkke 1 punktiga 7 ja ülemine haar staatori hamba 1 ülemise äärega. Pikkuse l_2 esimene osa on pool eelnimetatud kaarest. Pikkuse l_2 teine osa on võrdne kaare pikkusega, millele vastava nurga haarad on määratud staatori hammaste vahe suurusega. Pikkuse l_2 kolmas osa on pool kaarest, millele vastava nurga alumine haar on määratud staatori hamba 2 alumise äärega ja ülemine haar tõkke 1 punktiga 6. Pikkus l_3 on samuti pool eelnimetatud kaare pikkusest. Pikkused l_4, l_5 on mõlemad pool kaare pikkusest, millele vastava nurga alumine haar on määratud tõkke 2 punktiga 7 ja ülemine haar tõkke 2 punktiga 6.



Joonis 4.9 Tangentsiaalsete sildade takistuste määramine näite põhjal (q-telg)

Radiaalsete sildade puhul vastab igale magnetvoo tõkkele üks takistus. Radiaalse silla takistuse määramist vaadeldakse joonisel 4.10 toodud näite põhjal. Silla takistuse geometria määratakse vastava magnetvoo tõkke punktide 1 ja 2 järgi. Siinkohal on vajalik täpsustuseks lisada, et takistuse laius a on pool kogu takistuse ala laiuusest (teine pool jääb mootori alumise sektori sisse). Seda asjaolu on vajalik arvestada magnetahela lahendamisel.



Joonis 4.10 Radiaalse silla takistuse määramine näite põhjal (q-telg)

5. MAGNETAHELA LAHENDAMINE

Kahes eelnevas peatükis vaadeldud magnetahelate lahendamise peaeesmärgiks on leida staatori mähise induktiivsuse L_d ja L_q komponendid. Selles peatükis käsitletakse magnetomotoorjõu arvutamise põhimõtet, ahela lahendamist sõlmpingemeetodil, mittelineaarse ahela lahendi leidmise põhimõtet ja induktiivsuse arvutamist. Lisas 2 on esitatud *MATLAB* programmikood ahela koostamise automatiseerimiseks, mille põhimõte on toodud voodiagrammi abil lisas 1.

Magnetahela lahendusprotsess teostatakse eraldi d-teljele ja q-teljele, mis on kirjeldatav järgmiselt. Koostatakse sõlmpingemeetodile vastav võrrandisüsteem sisendparameetrite järgi, leitakse mittelineaarse ahela lõplik lahend, leitakse magnetvoo väärtus staatori igas hambas. Induktiivsuse väärtus on seejärel leitav hambale vastava aheldusvoo järgi. Staatori mähise kogu induktiivsus leitakse kõigi hammaste induktiivsuste summana. Kõik suurused vastavad seejuures mootori staatilisele olekule. Lisaks on vajalik märkida, et antud töös jäetakse välja d- ja q-teljele vastavate asendite vastastikune induktiivsus.

5.1 Magnetomotoorjõudude arvutamine

Järgnevalt leitakse hammastele vastavad *MMF*-i väärtused. Eelmistes peatükkides on *MMF*-id asetatud ahela lahendamise lihtsustamiseks õhupilusse, aga Kirchhoffi II seaduse kohaselt võib hambaga ühenduses olevate harude *MMF*-id viia üle vastava hamba harusse. Magnetomotoorjõud tekitab magnetahelas magnetvoo sarnaselt nagu elektromotoorjõud tekitab voolu elektriahelas. *MMF* ühes hambas sõltub seda hammast mõjutavatest keerdude (kontuuride) arvust ja nende voolu väärtusest [5]:

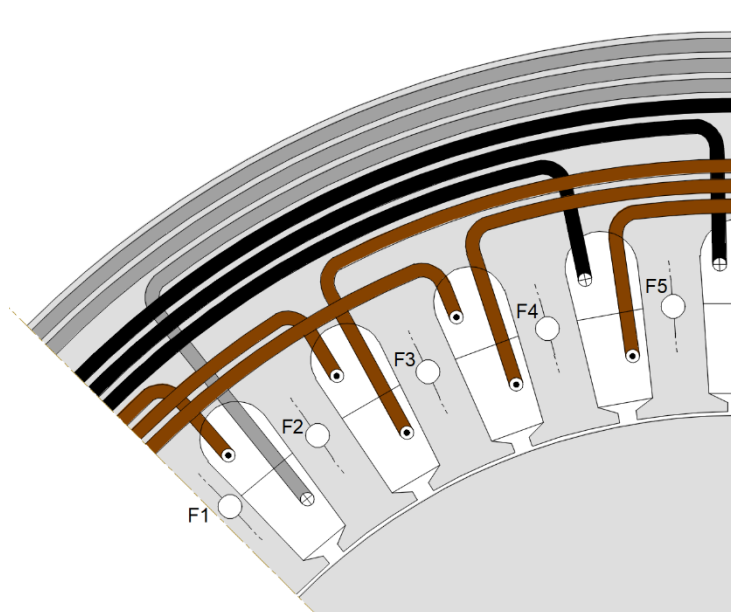
$$F_n = NT \cdot \sum_{m=1}^{PH} WP_{mn} \cdot i_m \quad (5.1)$$

kus F_n – hamba n *MMF*, A · keerdude arv,
 NT – mähise ühe pooli keerdude arv,
 PH – faaside arv,
 WP_{mn} – faasi m ja hamba n mähise jaotusviisi tegur,
 i_m – faasi m vool, A.

Antud töös arvestatakse ainult voolu põhiharmoonilise komponendi mõju. Lisaks on oluline märkida, et faaside vooludel peab olema valitud staatori asendile vastav algfaasi nurk ν_i (valem 1.1). Algfaasi nurga valik sõltub sellest, millise nurga suhtes staatorit vaadeldakse. Mähise poolt

tekitatav välja pooluse asukoht peab ühtima rootori d-teljega. Seejärel leitakse vaadeldava mootori sektori poolusele vastav magnetvoo suund hammastes (kas alt üles või ülevalt alla). Magnetvoo suuna järgi määratakse *MMF*-de suund.

Mähise jaotusviisi tegurit selgitatakse etteantud 10,3 kW mootori mähise põhjal (vt alapeatükk 2.1). Joonisel 5.1 on näidatud vaadeldavat mootori sektorit mõjutavad mähise poolid (kontuurid). Hamba *MMF*-i mõjutab üks pool siis, kui vastav hammas jääb pooli kontuuri sisse. Kontuuri poolt tekitatava magnetvoo suund hambas määratakse parema käe reeglga (vastavalt mineku või tuleku suunale). Kui poolusele vastav magnetvoo suund (eelnevalt leitud) ei ühti kontuuri poolt tekitatava magnetvoo suunaga, siis muudetakse teguri *WP* märk negatiivseks.



Joonis 5.1 Mähise jaotusviisi arvestamine *MMF*-de leidmisel (faaside värvid: A-pruun, B-must, C-hall)

5.2 Ahela lahendamine sõlmpingemeetodiga

Selles alapeatükis vaadeldakse sõlmpingemeetodi võrrandisüsteemi koostamist ja selle abil ahela lahendamist. Ahel lahendatakse eraldi d- ja q-teljele, aga lahendusprotsess on mõlemal juhul sarnane. Võrrandite arv vastab ahela sõlmede arvule miinus baassõlm (sõlm 0). Võrrandisüsteemi vasakul pool on juhtivuste maatriks, mille ridade ja tulpade arv on vastab sõlmede arvule. Juhtivuste maatriksi kordajate tulpmaatriksis on otsitavad sõlmede potentsiaalid. Võrrandisüsteemi paremal pool on sõlmede *MMF*-de mõjude summa tulpmaatriks. Võrrandisüsteem kujuneb välja järgmiselt:

$$\begin{bmatrix} G_{11} & -G_{12} & \cdots & -G_{1n} \\ -G_{21} & G_{22} & \cdots & -G_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -G_{m1} & -G_{m2} & \cdots & G_{nn} \end{bmatrix} \cdot \begin{bmatrix} U_{10} \\ U_{20} \\ \vdots \\ U_{n0} \end{bmatrix} = \begin{bmatrix} J_{11} \\ J_{22} \\ \vdots \\ J_{nn} \end{bmatrix} \quad (5.2)$$

kus G_{nn} – sõlme n juhtivus, $Wb/(A \cdot \text{keerdude arv})$,

G_{mn} – sõlmede m ja n vaheliste harude juhtivuste summa, $Wb/(A \cdot \text{keerdude arv})$,

U_{n0} – sõlme n ja sõlme 0 (baassõlme) potentsiaalide erinevus, A ,

J_{nn} – sõlmevoog sõlmele n , $(Wb \cdot \text{keerdude arv})$.

Selgituseks – sõlme juhtivus tähendab selle sõlme külge ühendatud harude magnetiliste juhtivuste summat ning sõlmevoog tähendab selle sõlme külge ühendatud harude *MMF*-de mõjude (haru *MMF*-i ja haru juhtivuse korrutis) summat. Kui sõlmede vahel ühendus puudub, võrdub vastav koht juhtivuste maatriksis nulliga.

Võrrandisüsteemi koostamise automatiseerimiseks luuakse *MATLAB* programmikood, mis on täispikkuses toodud lisas 2. Koodi põhimõtte on toodud lihtsustatud voodiagrammi abil lisas 1. Programmikoodi abil läbitakse ahela kõik sõlmed alustades esimesest sõlmest. Iga sõlmele määratakse järgnev informatsioon – sõlmelega ühenduses olevate harude juhtivused; lisateave järgmise sõlme (või sõlmede) asukoha kohta; *MMF*-i väärtus juhul, kui sõlmelega ühenduses olevas harus on *MMF*; muu vajalik info. Saadud informatsiooni põhjal leitakse juhtivuste maatriks ja sõlmevoogude maatriks. Täpsustus – tärniga on märgitud erinevused d- ja q-telje vahel.

Eelnevalt nimetatud programmikoodi abil leitakse d- ja q-telje ahelatele kõigi takistuste geometriad, millele lisaks on vajalik arvestada lekkevoogu mõju. Seda arvestatakse ainult õhu takistuste juures, tingituna suuresti õhupilu geomeetrisest omapärast (suur lekkevoogu osatähtsus). Lekkevoogu arvestatakse fikseeritud koefitsiendiga λ (Hopkinsoni koefitsient [24]) iga takistuse kohta (takistus jagatakse koefitsiendiga λ). Koefitsient on kirjeldatav kui kogu voo ja kasuliku voo suhe. λ leitakse eksperimentaalsel teel õhupilu takistuste järgi. Selleks koostatakse joonisel 3.7 kujutatud ahel ainult õhupilu takistustega (lineaarne ahel). Esiteks leitakse ahela „kasulik“ induktiivsus lahendades eelnimetatud ahela (ilma lekkevoogu mõjuta). Selleks asendatakse *MMF*

hambas n (valem 5.1) keerdude arvu ruuduga $F_n = (NT \cdot \sum_{m=1}^{PH} WP_{m,n})^2$. Teiseks leitakse ahela „kogu“ induktiivsus (koos lekkevoe mõjuga) kasutades tarkvara *FEMM*. Koefitsiendi λ saamiseks tuleb leitud induktiivsuste suhtest $\left(\frac{\text{"kogu" induktiivsus}}{\text{"kasulik" induktiivsus}}\right)$ võtta ruutjuur. Viimane on vajalik, sest induktiivsused on leitavad keerdude arvu ruudu abil. Lõpptulemusena saadud koefitsiendi väärtus on ligikaudu $\lambda = 1,12$.

Järgmiseks lahendatakse võrrandisüsteem 5.2, mille tulemusena saadakse sõlmede potentsiaalid baassõlme suhtes. Terasse takistusi sisaldava haru m voo väärtus leitakse järgnevalt:

$$\Phi_m = \frac{U_{n0} - U_{(n+1)0}}{R_m} \quad (5.3)$$

kus Φ_m – haru m magnetvoog, Wb,

R_m – haru m takistuste summa, $(A \cdot \text{keerdude arv})/\text{Wb}$.

Täpsustus – sõlmed n ja $n+1$ peavad olema haruga m ühendatud. Lisaks peab positiivse tulemuse jaoks haru m voog olema suunatud sõlmest n sõlme $n+1$.

Peatüki algul sõnastatud eesmärgi põhjal on vaja leida magnetvoogude väärtused staatori igas hambas. Selleks leitakse vood õhupilu takistust sisaldavates harudes, sest viimased on alati ühenduses hammastega. Kirchhoffi I seaduse kohaselt liidetakse nende harude vood iga hamba kohta. Õhupilu takistust sisaldava haru m voo väärtus q -telje korral leitakse järgnevalt (baassõlm asub staatoris):

$$\Phi_m = \frac{F_m + U_{n0}}{R_m} \quad (5.4)$$

Täpsustus – sõlm n peab olema haruga m ühendatud. Lisaks peab haru m voog ühtima *MMF-i* F_m suunaga ning olema suunatud väljapoole sõlmest n (vastasel korral muutub vastav märk).

D-telje korral jääb õhupilu takistust sisaldav haru m alati kahe sõlme vahele, mistõttu haru voo väärtus leitakse järgnevalt:

$$\Phi_m = \frac{F_m + U_{n0} - U_{(n+1)0}}{R_m} \quad (5.5)$$

Täpsustus – sõlmed n ja $n+1$ peavad olema haruga m ühendatud. Lisaks peab haru m voog ühtima *MMF-i* F_m suunaga ning olema suunatud sõlmest n sõlme $n+1$ (vastasel korral muutub vastav märk).

Nimetatud magnetvoogude leidmiseks on vajalik eelnevalt kõigile terase takistustele anda ette suhteline magnetiline läbitavus μ_r , mis on vajalik nende takistuse (ja juhtivuse) arvutamiseks (vt valem 3.2 ja 3.3). Realse terase magnetiline läbitavus sõltub aga mittelineaarselt

magnetvootihedusest. Seetõttu on d- ja q-telje ahelate lõplike lahendite leidmiseks vajalik iteratiivse protsessiga lahendusmeetod, mida vaadeldakse järgnevas alapeatükis.

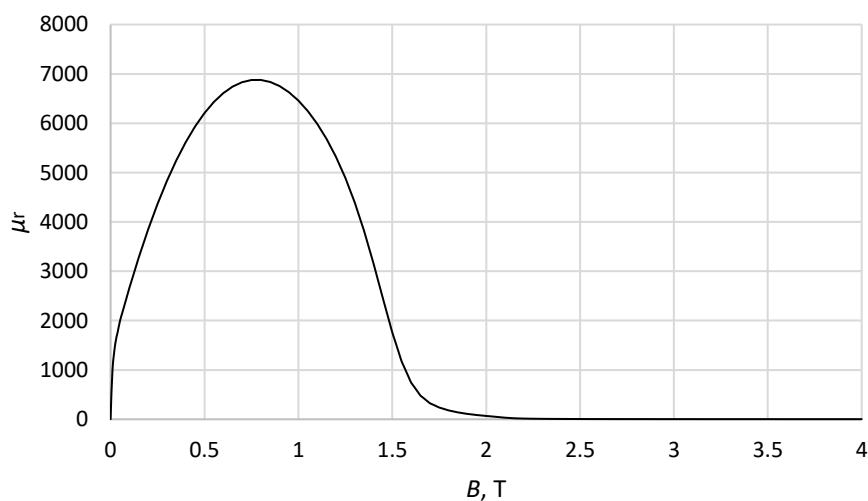
5.3 Mittelineaarse ahela lahendi leidmine

Antud töös käsitletavat magnetahelat sisaldavad mittelineaarseid komponente. Järgnevalt vaadeldakse sellise mittelineaarse ahela lahendi leidmist iteratiivse protsessi abil. Mittelineaarseteks komponentideks on kõik ahela terase magnetilised takistused. See tuleneb sellest, et terase takistus sõltub suhtelisest magnetilisest läbitavusest μ_r (vt valem 3.2). Antud töös kasutatava etteantud 10,3 kW mootori näitel on terase tüübiks M-45. Terase M-45 suhtelise magnetilise läbitavuse μ_r mittelineaarne sõltuvus magnetvootihedusest B on toodud joonisel 5.2. Magnetvootihedus avaldub magnetvoo ja selle poolt läbitava pindala suhena:

$$B = \frac{\Phi}{A} \quad (5.6)$$

kus B – magnetvootihedus, T,

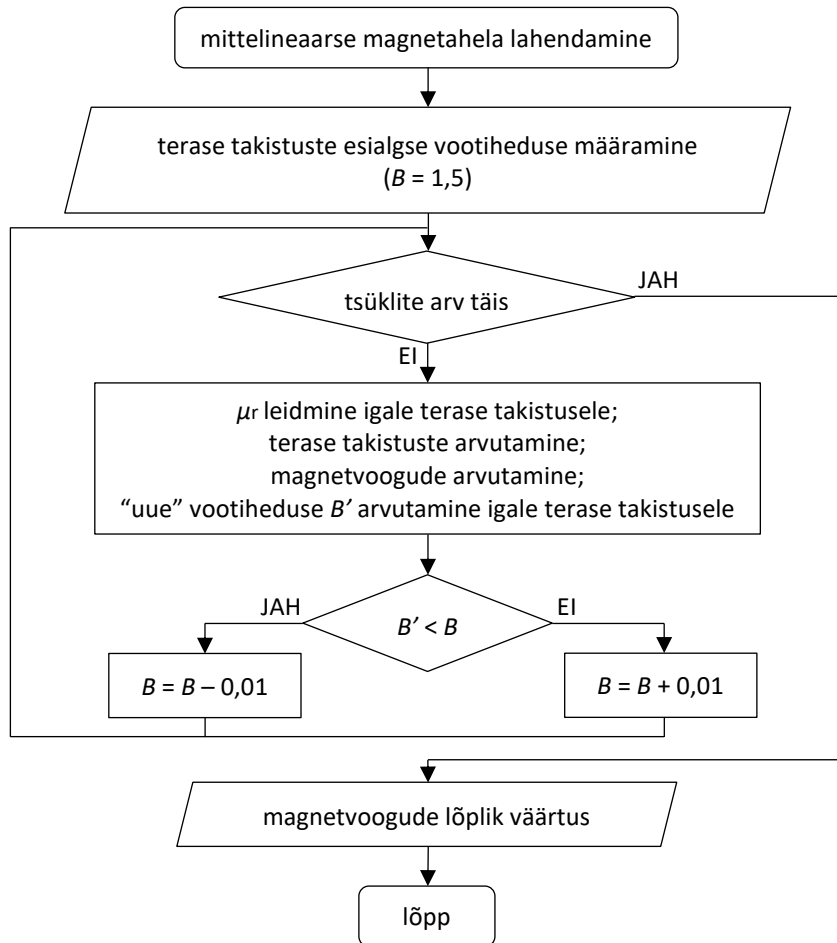
Φ – magnetvoog, Wb.



Joonis 5.2 Terase M-45 suhtelise magnetilise läbitavuse μ_r sõltuvus magnetvootihedusest B

Mittelineaarse ahela lahendi leidmiseks on vajalik leida ahela igale terase takistusele sobilik μ_r . Selleks kasutatakse iteratiivset protsessi, mis sisaldub lisa 2 toodud programmikoodis. Selle tööpõhimõtte on lihtsustatud voodiagrammi abil kujutatud joonisel 5.3. Takistuste esialgseks arvutamiseks antakse kõigi takistuste jaoks ette tinglik vootihedus $B = 1,5$ T. Järgnevalt teostatakse kindel arv iteratsioone, milles sisalduvad järgnevad tegevused. Vootiheduse B järgi leitakse igale takistusele vastav μ_r . Leitud μ_r väärtuste abil arvutatakse kõik terase takistused (takistuste geometria on eelnevalt fikseeritud). Seejärel arvutatakse välja ahela terase takistusi sisaldavate

harude magnetvoog. Nende voogude põhjal arvutatakse igale takistusele „uus“ vootihedus B' . Kui B' on väiksem kui eelnev vootihedus B , siis parandatakse vootihedust $B - 0,01$ võrra. Vastasel korral parandatakse vootihedust $B + 0,01$ võrra (suurus $0,01$ valitakse piisava täpsuse saavutamiseks). Järgnev iteratsioon teostatakse parendatud B väärtustega. Kui iteratsioonide arv on täidetud väljastatakse ahela lõpliku lahendina viimased magnetvoogude väärtused.



Joonis 5.3 Lihtsustatud voodiagramm mittelineaarse magnetahela lahendamiseks

5.4 Induktiivsuse leidmine

Eelnevas alapeatükis leiti magnetvoo väärtused vaadeldava mootori sektori igas hambas rootori d- ja q-teljele vastavatele asenditele. Seejuures jäetakse välja lekkevoog staatori uuretes. Mähise induktiivsus leitakse samuti eraldi igale hambale (s.t. sellele hambale vastavatele mähise poolidele), mis avaldub järgmiselt:

$$L_n = \frac{\Psi_n}{i_{\text{avg}}} \cdot k_w \quad (5.7)$$

kus L_n – hambale n vastav induktiivsus, H,
 Ψ_n – hambale n vastav aheldusvoog, (Wb · keerdude arv),
 i_{avg} – faaside voolude hetkväärtuste aritmeetiline keskmine, A,
 k_w – mähkimistegur.

Faaside voolude keskmist väärtust i_{avg} kasutatakse lahenduse lihtsustamise eesmärgil. See väärtus omistatakse kõikidele faasidele ($i_{\text{avg}} = i_A = i_B = i_C$) ja see jääb samaks nii d-telje kui ka q-telje korral. Kogu mootori mähise induktiivsuse leidmiseks liidetakse leitud staatori hammastele vastavad väärtused ja korrutatakse tulemus kaheksaga (ahel vastab 1/8 kogu mootorist).

Hambale n vastav aheldusvoog Ψ_n avaldub kui hamba n magnetvoo Φ_n ja selle voo poolt aheldatud mähise poolide kontuuride arvu korrutis:

$$\Psi_n = \Phi_n \cdot NT \cdot \sum_{m=1}^{PH} WP_{mn} \quad (5.8)$$

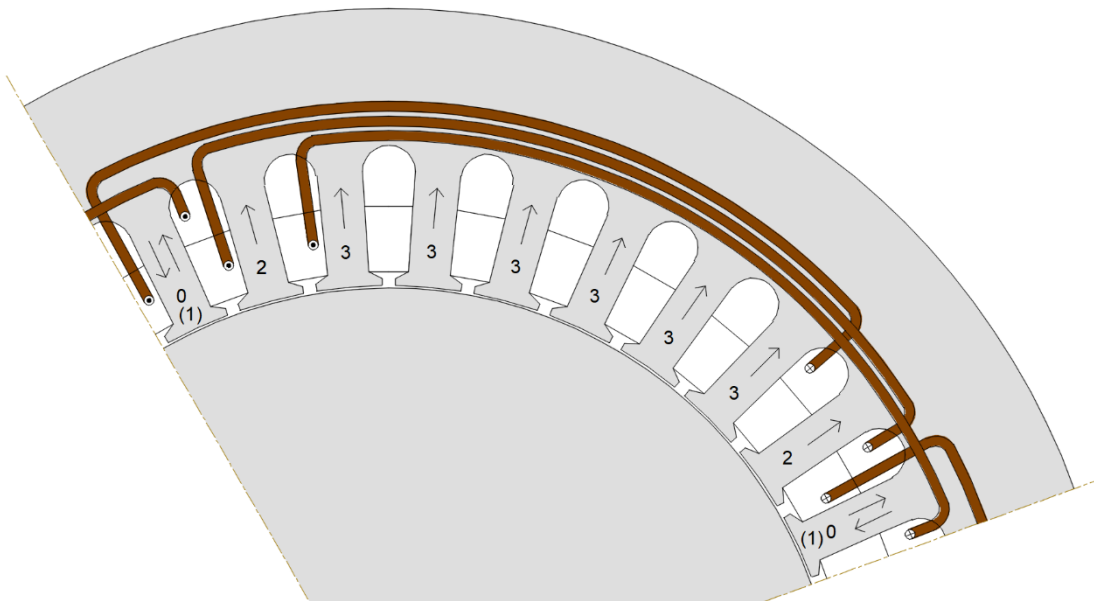
Jaotusviisi teguri WP_{mn} (faas m , hammas n) leidmist on kirjeldatud alapeatükis 5.1. Erinevus on siinkohal teguri märgis (pluss või miinus) – märk võetakse vastupidiseks juhul kui faasi m vool on negatiivne.

Mähkimistegur k_w on siinkohal lihtsustatud tegur mähkimisviisist tuleneva mõju arvestamiseks. Mähkimistegur on üldjuhul defineeritav kui vaadeldava mähise poolt aheldatava magnetvoo ja võrdlusmähise poolt aheldatava magnetvoo suhe [29]. Võrdlusmähis on seejuures täis sammuga, erinevate pooluste mähise poolid ei kattu ja selle mähkimistegur on võrdne ühega. Antud lihtsustatud mähkimisteguriga k_w arvestatakse ainult erinevate pooluste poolide kattumise mõju. Kahe erineva pooluse poolide kattumisel on nende summaarne aheldatav voog null (vastava hamba kohta). Mähkimistegur k_w avaldub järgmiselt:

$$k_w = \frac{\text{vaadeldava mähise poolt aheldatav magnetvoog}}{\text{võrdlusmähise poolt aheldatav magnetvoog}} \quad (5.9)$$

Täpsustus – magnetvoog võetakse kõikides hammastes võrdseks.

Teguri leidmist on näidatud etteantud 10,3 kW mootori mähise (vt alapeatükk 2.1) põhjal joonisel 5.4. Vaadeldakse ainult ühe faasi mähist, sest teiste faaside korral on põhimõtte sama. Numbritega on toodud hambale vastav mähise poolide arv vaadeldava mähise korral. Sulgudes olev number näitab võrdlusmähisele vastavat poolide arvu. Antud näite (etteantud mootori) korral on mähkimisteguri väärtus ligikaudu $k_w = 0,917$.



Joonis 5.4 Mähkimisteguri k_w leidmise põhimõtte

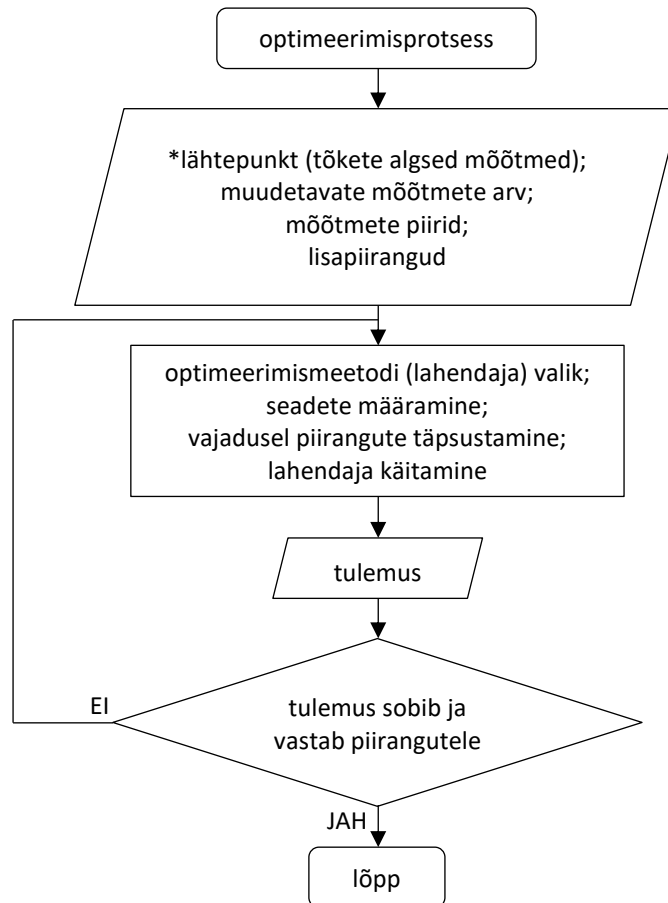
6. ROOTORI OPTIMEERIMINE

Antud töös optimeeritakse sünkroon-reluktantsmootori rootorit eelnevas peatükis leitud induktiivsuse L_d ja L_q komponentide maksimaalse erinevuse (suhte L_d/L_q) saavutamiseks. Seejuures jäetakse vaatluse alt välja teistele mootori karakteristikutele avaldatav mõju. Optimaalse mootori projekteerimiseks on üldjuhul vajalik kasutada mitme-eesmärgilist optimeerimist, mida antud töös ei käsitleta. Selles peatükis kirjeldatakse optimeerimise põhimõtet ning vaadeldakse optimeerimismeetodi valikut etteantud 10,3 kW mootori näite põhjal.

Maksimeerides induktiivsuste L_d , L_q erinevust saavutatakse sellega ühtlasi maksimaalne keskmine õhupilu moment (vt valem 1.7). Mootori momenti analüüsis on eelkõige vajalik saavutada maksimaalne keskmine moment ja minimaalne pulsatsioonimoment [10]. Kuna antud töös maksimeeritakse keskmist õhupilu momenti, siis on tähtis vaadelda sellest tulenevat mõju pulsatsioonimomendile. Selleks vaadeldakse antud peatükis lisaks etteantud 10,3 kW mootori pulsatsioonimomendi muutust esialgse ja optimeeritud rootori vahel.

6.1 Optimeerimise põhimõte

Optimeerimisprobleemi defineerimiseks peavad eelnevalt olema fikseeritud mootori peamised mõõtmed ning staatori ja selle mähise ehitus (vt alapeatükk 2.1). Lisaks on vajalik fikseerida magnetvoo tõkete arv ja nende mõõtmete piirid. Enamike optimeerimismeetodite korral on võimalik määrata lisaks lähtepunkt – kindlad tõkete mõõtmed, mis jäävad määratud piiridesse. Muudetavateks suurusteks on optimeerimise käigus magnetvoo tõkete mõõtmed, mis on toodud tabelis 2.6. Sõltuvalt probleemist võivad mõningad mõõtmed olla ka fikseeritud. Lõplikuks lahendiks on magnetvoo tõkete kindla asendi määravad mõõtmed, millele vastab suurim suhe L_d/L_q ja kehtivad piirangud. Lahendi sobivust on vaja siiski hinnata lähteülesande suhtes. Vajadusel valitakse teine lahendaja või parandatakse lahendaja seadeid või täpsustatakse piiranguid. Optimeerimisprotsess (optimeerimisprobleemi lahendamine) on voodiagrammi abil toodud joonisel 6.1.



Joonis 6.1 Rootori optimeerimisprotsess

Lisapiiranguteks võivad olla nii lineaarsed kui ka mittelineaarsed piirangufunktsioonid. Piirangufunktsioonidega võib määrata näiteks mehaanilisi piiranguid, kindlate alade mõõtmel piiranguid, külgede omavahelist asetust ja teisi probleemi eripärast tulenevaid piiranguid. Siinkohal on vajalik märkida, et sõltuvalt probleemi tüübist võib tekkida vajadus piirata kahe tõkke vahele jäävate alade külgede asetust. Nimetatud küljed peavad olema omavahel piisavalt paralleelsed, mis on vajalik püsiva magnetvootiheduse saavutamiseks nendes alades. See tagab terase efektiivse kasutuse [7].

Lahendusmeetodi valik sõltub suuresti optimeerimisprobleemi eripärast ja selle sisendparameetritest (tõkete arv, muudetavate mõõtmel arv, mõõtmel piirangud, mõõtmel omavaheline seos, lisapiirangud). Iga meetodi (lahendaja) seadistus sõltub vastavalt vajadusele ja nõuab üldjuhul meetodi lähemat tundmist. Antud põhjustel ei vaadelda süvitsi erinevate lahendusmeetodite kasutamise sobivust ja võimalusi.

6.2 Etteantud 10,3 kW mootori optimeerimine

Järgnevalt vaadeldakse rootori optimeerimist etteantud 10,3 kW mootori näitel kahe juhtumi põhjal. Esimesel juhul fikseeritakse radiaalsete ja tangentsiaalsete sildade laiused lähtepunkti järgi. Teisel juhul fikseeritakse radiaalsete sildade laiused parendatud väärtuste järgi ja tangentsiaalsete sildade laiused on muudetavad suurused. Mõlemal juhul on lähtepunktiks on etteantud mootori esialgsed tõkete mõõtmed (tabel 2.6). Esialgsed muutuvate suuruste (muutujate) piirid määratakse hinnanguliselt selliselt, et mitte piirata võimalikke tulemusi, aga siiski hoida arvutusaeg piisavalt väike. Peamiseks raskuseks on muutujate piiride seadmine selliselt, et kõik tõkked jääksid vastava ala sisse ja ei kattuks omavahel. Selle saavutamiseks luuakse (mittelineaarne) piirangufunktsioon (lisa L2.3). Lisaks sellele kasutatakse piirangufunktsiooni kahe tõkke vahele jäävate alade külgede piisava paralleelsuse saavutamiseks.

Järgnevalt on vajalik leida antud optimeerimisprobleemile sobilik lahendusmeetod. Probleemi lahendamiseks kasutatakse tarkvara *MathWorks MATLAB* poolt pakutavaid tööriistu. Need kasutavad üldjuhul erinevaid lahendusmeetodeid (lahendajaid) funktsiooni miinimumi leidmiseks arvestades seejuures lisatud piiranguid. Enne täpsema lahendusmeetodi valikut on vajalik täpsustada kas otsitakse lokaalset või globaalset ekstreemumi. Kuna funktsiooni omaduste kohta info puudub ja muutujate erinevaid kombinatsioone on palju, siis kasutatakse globaalset meetodit. Samuti on raskesti hinnatav kas funktsioon on kumer/nõgus või omab kriitilisi punkte. Seetõttu valitakse lahendajapõhine (*solver-based*) lähenemine ja mittelineaarne probleem [30].

Sobiva lahendaja leidmiseks viiakse läbi optimeerimisprotsess kõigi peamiste lahendusmeetoditega. Saadud tulemuste põhjal valitakse sobivaimaks meetodiks *genetic algorithm*. Tabelis 6.1 on toodud lühidalt ülejäänud lahendajate mittedobivuse põhjused. *Genetic algorithm* sobib antud probleemi lahendamiseks, kuna selle meetodiga liigutakse järkjärgult globaalse ekstreemumi poole kasutades eelnevate sammude punktide asukohti. Nimetatud meetodit on laialdaselt kasutatud kirjanduses sarnaste probleemide lahendamiseks (näiteks [31, 32, 33, 5]). Samuti sobib *genetic algorithm* mitme-eesmärgiliste probleemide lahendamiseks.

Tabel 6.1 Lahendajate mittedobivuse põhjused

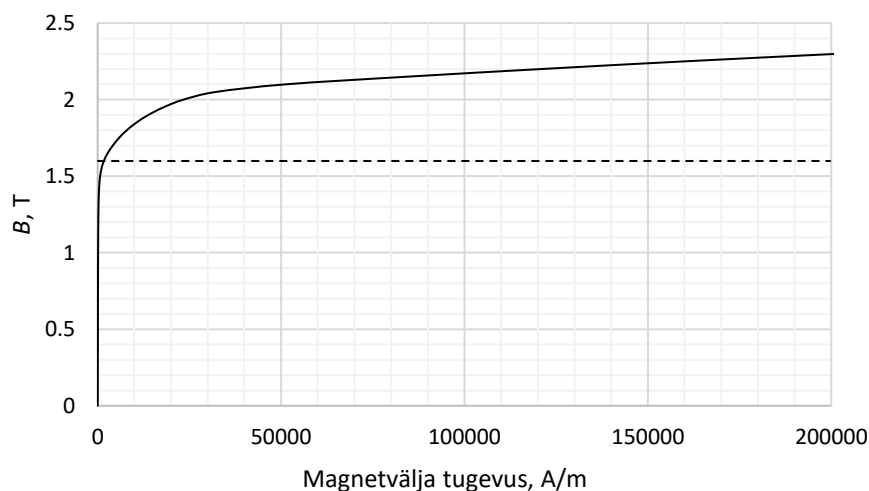
Lahendaja	Põhjendus
<i>MultiStart/GlobalSearch</i>	Mõlema lahendaja töö põhineb juhuslike punktide ümbruses lokaalse ekstreemumi leidmisel [34]. Ei ole efektiivne ja täpne antud probleemi lahendamiseks (muutujate arv suur – palju erinevaid kombinatsioone). Sobivat tulemust ei saadud.
<i>patternsearch</i>	Lahendaja poolt juhuslikult genereeritavad punktid ületavad seatud piiranguid. Sobivat tulemust ei saadud.
<i>particleswarm</i>	Ei arvesta piirangufunktsiooni, lähtepunkti. Sobivat tulemust ei saadud.
<i>surrogateopt</i>	Ei arvesta piirangufunktsiooni. Lahendaja poolt juhuslikult genereeritavad punktid ületavad seatud piiranguid. Sobivat tulemust ei saadud.

Optimeerimisprotsess viiakse läbi *genetic algorithm* lahendajaga kahe juhtumi jaoks. Juhtumi 1 korral on radiaalsete (mõõde B_2) ja tangentsiaalsete (mõõde B_4) sildade laiused määratud lähtepunkti järgi ja nende laius on fikseeritud (tabel 6.2, juhtum 1). Sildade laiused on piiratud eelkõige rootori mehaanilise tugevusega. Ideaalsel juhul radiaalsed sillad puuduvad. Optimaalsete laiuste leidmiseks on vajalikud materjalist sõltuvad mehaanilised tugevusarvutused, mida antud töös ei käsitleta. Juhtumi 2 korral parendatakse sildade laiusi hinnanguliselt kirjanduses toodud näidete [7, 31, 35] põhjal. Seejuures fikseeritakse radiaalsete sildade laiused ja tükke 1 tangentsiaalse silla laius (tabel 6.2, juhtum 2). Muutujateks on tükete 2...4 tangentsiaalsete sildade laiused. Mõlema juhtumi korral on ülejäänud tükete mõõtmed muutujateks.

Tabel 6.2 Sildade laiused kahe vaadeldava juhtumi jaoks

Juhtum	Tükke nr	B_2 , mm	B_4 , mm
1	1	2,5	1,0
	2	2,5	1,0
	3	2,5	1,0
	4	2,5	1,0
2	1	2,5	1,0
	2	2,0	0,9...1,0
	3	1,5	0,8...1,0
	4	1,0	0,7...1,0

Terase M-45 magnetilise küllastuse tase valitakse vastavaks ligikaudu 1,6 T magnetvootihedusele, millele vastav mähise voolu amplituudväärtus on $I_m = 5$ A. Teras magnetvootiheduse B sõltuvus magnetvälja tugevusest on toodud joonisel 6.2. Sellelt on näha, et kõvera (alumine) lineaarne osa jääb allapoole väärtusest ligikaudu 1,6 T. Seega sellel tasemel on kasutatud ära enamuse materjali efektiivsest magnetilisest omadusest. Üle selle taseme väheneb oluliselt välise magnetvälja tugevuse mõju magnetvootiheduse väärtusele B . Lisaks on oluline märkida, et suhe L_d/L_q sõltub suuresti magnetmaterjali küllastuse tasemest (mähise voolu väärtusest) [36, 37, 38, 6].



Joonis 6.2 Teras M-45 magnetvootiheduse B sõltuvus magnetvälja tugevusest

Optimeerimisprobleemi lahendaja seadete määramisel lähtutakse nende üldkirjeldusest [39] ja mõjust tulemustele. Antud probleemi korral on parima tulemuse saavutamiseks vajalik protsessi korrata. Seejuures igal järgneval korral võetakse lähtepunktiks eelnev tulemus ja arvutusaja vähendamiseks kitsendatakse hinnanguliselt muutujate piire.

Lahenduse tulemusena saadud magnetvoo tõkete mõõtmed on juhtumi 1 korral toodud tabelis 6.3 ja juhtumi 2 korral tabelis 6.4 (mõõtmete selgitus joonisel 2.2). Tõkete asetustele vastavad induktiivsuste väärtused ja suhted on toodud tabelis 6.5 (mudel – magnetahel). Võrdluseks on toodud induktiivsused *FEMM* mudeli põhjal (mudel – *FEMM*). See põhineb lõplike elementide meetodil, millega saavutatakse mudeli suurem täpsus. Täpsustus – toodud induktiivsuste väärtused vastavad magnetostaatilisele mudelile (*FEMM* lahenduse tulemuste reaalosa). Lisaks on tabeli viimases tulbas toodud induktiivsuse suhtele vastav maksimaalne võimsustegur. Katselisel teel mõõdetud etteantud (esialgse) mootori võimsustegur on 0,6 (mõõteprotsessi antud töös ei käsitleta). See on lähedane antud juhul *FEMM* mudeli maksimaalsele võimsustegurile 0,59.

Tabel 6.3 Optimeerimisel saadud magnetvoo tõkete mõõtmed juhtum 1 korral

Nr	B1, mm	B2, mm	B3, mm	B4, mm	B5, mm	B6, °	B7, °	B8, °
1	5,904	2,5	20,122	1,0	5,010	4,313	3,937	70,680
2	17,775	2,5	15,525	1,0	5,769	2,670	11,762	70,167
3	29,064	2,5	12,481	1,0	4,949	2,903	19,906	68,011
4	38,507	2,5	7,892	1,0	5,187	4,010	28,808	63,479

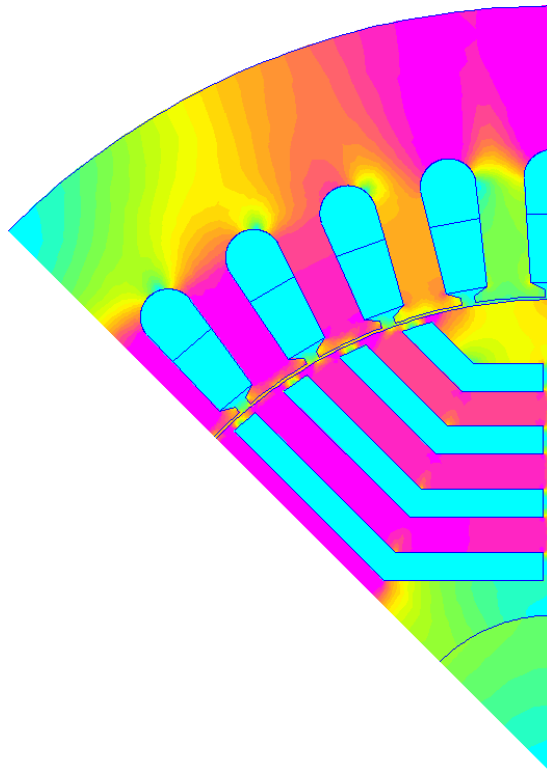
Tabel 6.4 Optimeerimisel saadud magnetvoo tõkete mõõtmed juhtum 2 korral

Nr	B1, mm	B2, mm	B3, mm	B4, mm	B5, mm	B6, °	B7, °	B8, °
1	5,882	2,5	20,219	1,0	6,379	3,056	3,104	72,046
2	18,723	2,0	15,80	0,902	6,373	2,984	11,842	69,504
3	29,607	1,5	12,355	0,808	5,416	2,915	19,733	72,854
4	39,416	1,0	8,963	0,708	5,184	4,688	29,303	64,105

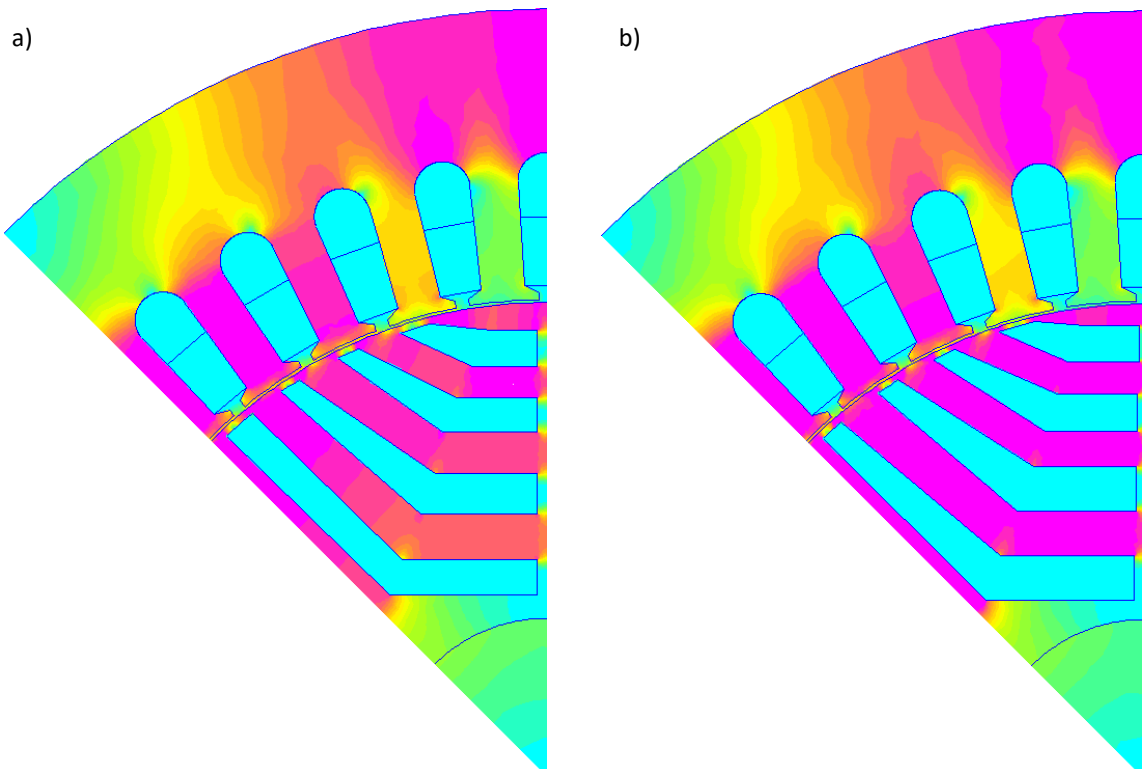
Tabel 6.5 Optimeerimisel saadud induktiivsused (magnetostaatilised väärtused) ja vastavad võimsustegurid

Juhtum	Mudel	L_d , H	L_q , H	L_d/L_q	Maksimaalne võimsustegur
esialgne rootor	magnetahel	1,783	0,431	4,136	0,61
	<i>FEMM</i>	1,688	0,437	3,866	0,59
1	magnetahel	1,860	0,386	4,817	0,66
	<i>FEMM</i>	1,710	0,417	4,101	0,61
2	magnetahel	1,806	0,310	5,831	0,71
	<i>FEMM</i>	1,70	0,327	5,195	0,68

Viimaks tuuakse illustratiivselt tõkete paigutus ja magnetvootiheduse jaotus. Esialgne rootor on toodud joonisel 6.3. Sinine värvus vastab vähimale ja roosa värvus suurimale magnetvootihedusele. Juhtumi 1 lõpliku tulemusena saadud tõkete paigutus on toodud joonisel 6.4, a. Juhtumi 2 lõpliku tulemusena saadud tõkete paigutus on toodud joonisel 6.4, b. Joonistelt on magnetvootiheduse põhjal näha, et optimeeritud tõkete asetuse korral on rootori terase pind kasutatud efektiivsemalt ja magnetvootihedus jaotunud ühtlasemalt.



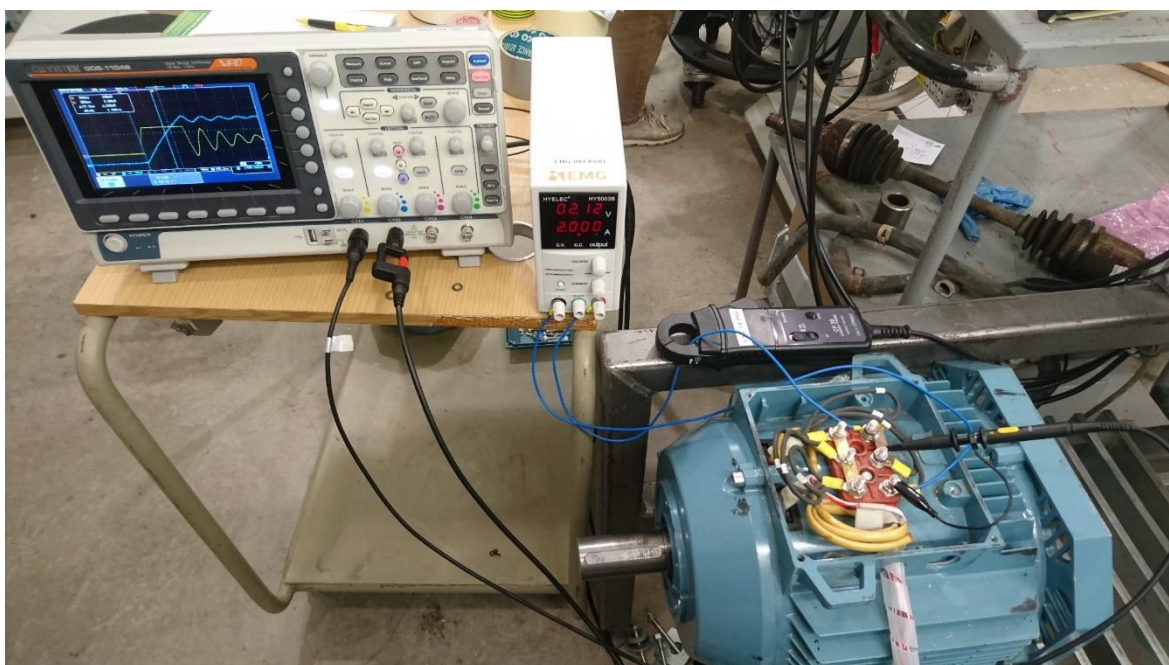
Joonis 6.3 Esialgse rootori tõkete paigutus ja magnetvootiheduse jaotus



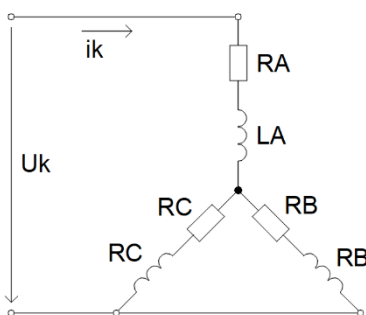
Joonis 6.4 Optimeerimise tulemusena saadud rootori tõkete paigutus ja magnetvootiheduse jaotus: juhtum 1 (a), juhtum 2 (b)

6.3 Induktiivsuste katseline mõõtmine

FEMM ja magnetahelate mudelite valideerimiseks mõõdetakse etteantud (esialgse) 10,3 kW mootori mähise d- ja q-telje induktiivsused katselisel teel. Mõõtmised viiakse läbi vastavalt kirjanduses toodud juhendile [40]. Vaadeldakse staatilist olekut ja kasutatakse alalisvoolu toiteallikat. Mootori mähise väljaviikudelt mõõdetakse igale faasile vastavad takistused $R_A = R_B = R_C = 0,7 \Omega$. Enne induktiivsuste mõõtmist leitakse ja märgitakse mootori võlli d- ja q-teljele vastavad asendid. D-telje leidmiseks ühendatakse faasi A väljaviik toiteallika positiivse klemmiga ja faasid B, C toiteallika negatiivse klemmiga. Q-telje leidmiseks ühendatakse faasi B väljaviik toiteallika positiivse klemmiga ja faas C toiteallika negatiivse klemmiga (faas A jäetakse ühendamata). Voolu väärtuseks valitakse 2 A (ligikaudu 10% nimivoolust), millele vastav pinge leitakse mähise takistuste järgi. Ülevaade mõõtmisest on toodud joonisel 6.5.



Joonis 6.5 Katseline induktiivsuste mõõtmine



Joonis 6.6 Mootori mähise väljaviikude ühendusviis mõõteahelas

Induktiivsuste mõõtmisel kasutatav ühendusviis on toodud joonisel 6.6. Induktiivsused L_d , L_q leitakse voolu i_k tõusu ajakonstandi τ järgi. Vool i_k avaldub järgmiselt:

$$i_k = \frac{U_k}{R_k} \cdot \left(1 - e^{-\frac{t}{\tau}}\right) \quad (6.1)$$

kus i_k – kogu mõõteahela voolu hetkväärtus, A,
 U_k – kogu mõõteahela mähise väljaviikudele vastav pinge, V,
 R_k – kogu mõõteahela takistus, Ω ,
 τ – ajakonstant, s.

Ajakonstant τ avaldub järgmiselt:

$$\tau = \frac{L_k}{R_k} \quad (6.2)$$

kus L_k – kogu mõõteahela induktiivsus, H.

L_d -le vastava ajakonstandi leidmiseks fikseeritakse võll d-teljele vastavale asendile. Järgmiseks antakse mootorile toiteallikast pinge U_k ning ostsilloskoobi abil leitakse tulemusena saadud voolu kõverale (tõusule) vastav ajakonstant τ . Induktiivsus avaldub järgmiselt:

$$L_d = \frac{2}{3} \cdot \tau \cdot R_k \quad (6.3)$$

L_q -le vastav ajakonstant τ leitakse analoogselt, aga võll fikseeritakse q-teljele vastavale asendile. Induktiivsus avaldub järgmiselt:

$$L_q = \frac{2}{3} \cdot \tau \cdot R_k \quad (6.4)$$

Mõõtmiste tulemused on toodud tabelis 6.6. Võrdluseks on toodud induktiivsused *FEMM* mudeli põhjal. Antud juhul on võetud *FEMM* mudelis voolu sageduseks 50 Hz. Induktiivsuste väärtused leitakse seejuures lahenduse tulemuste imaginaarosa järgi. Erinevus mõõdetud ja *FEMM* mudeli vahel on d-telje korral 8,0% ja q-telje korral 2,9%. Tulemuste erinevus võib tuleneda suuresti mootori konstruktsiooni erinevustest võrreldes mudeliga ja mõõteseadmete ebatäpsusest.

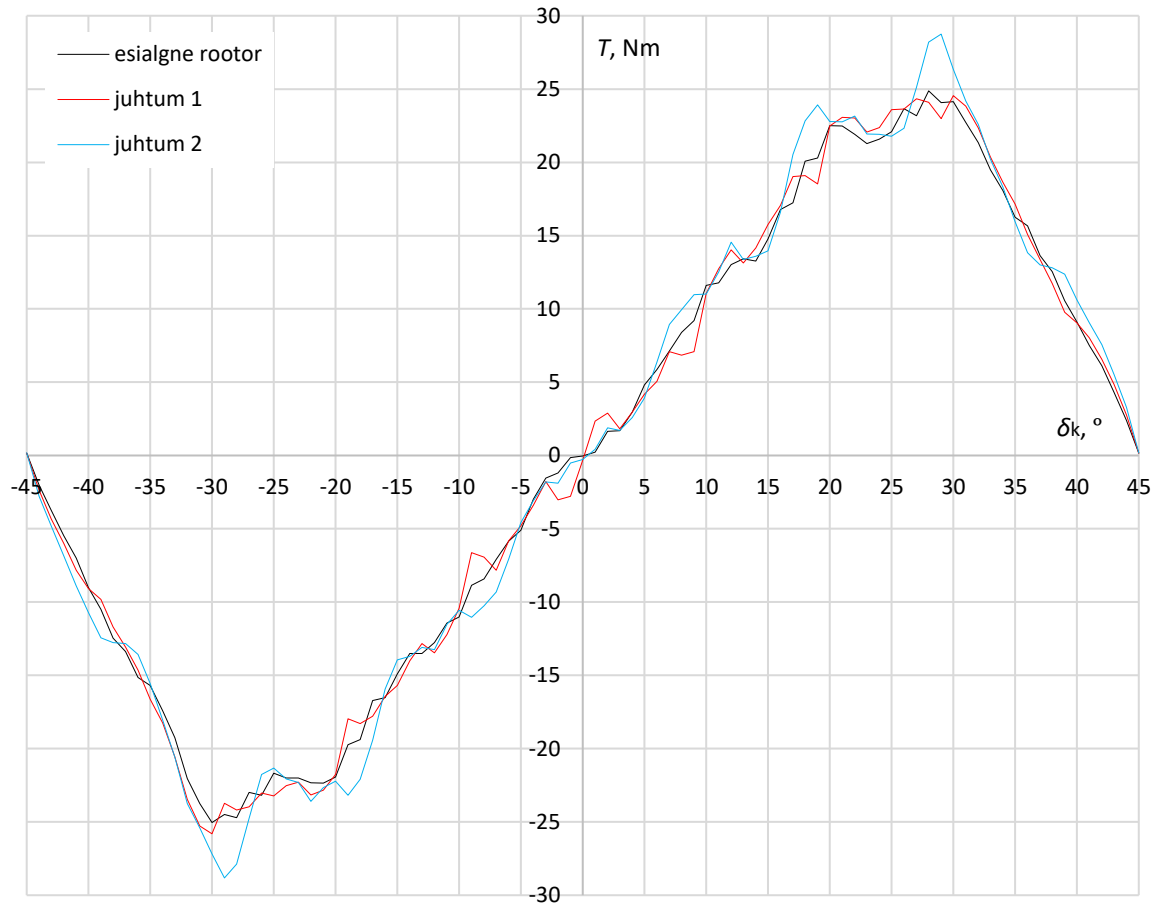
Tabel 6.6 Katselise mõõtmise ja vastava *FEMM* mudeli induktiivsused

Tulemus	L_d , H	L_q , H
katseline mõõtmine	0,0541	0,0277
<i>FEMM</i>	0,0501	0,0285

6.4 Pulsatsioonimoment

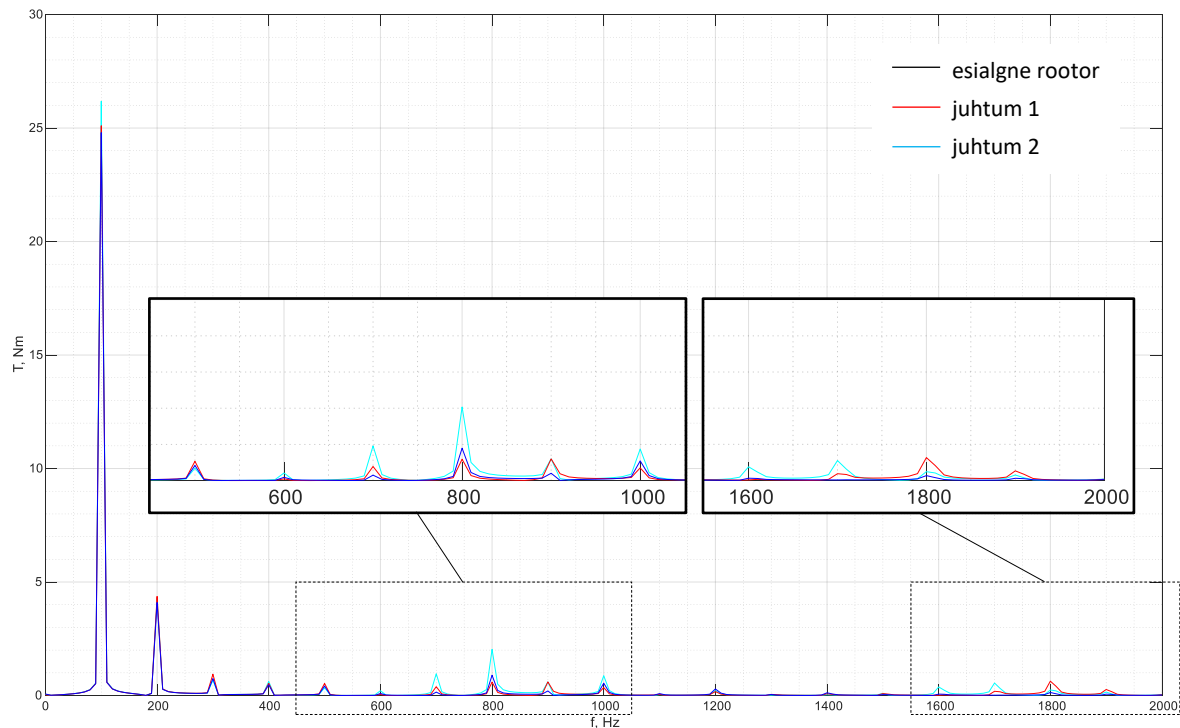
Antud töös maksimeeritakse induktiivsuste L_d , L_q erinevust, millega saavutatakse maksimaalne keskmine õhupilu moment T_{avg} (valem 1.7). Üldjuhul kaasneb keskmise momendi T_{avg} maksimeerimisega pulsatsioonimomendi suurenemine [6]. Pulsatsioonimoment suurendab mootori vibratsiooni, müra ja vähendab efektiivsust. Seega on mootori projekteerimisel tähtis ühtlasi minimeerida pulsatsioonimomenti [10]. See nõuab mitme-eesmärgilist optimeerimist, mida antud töös ei käsitleta. Selles alapeatükis vaadeldakse etteantud 10,3 kW mootori rootori optimeerimisega kaasnevat pulsatsioonimomendi muutust.

Pulsatsioonimomendi muutuse hindamiseks võetakse üles rootorile mõjuva staatilise oleku momendi T väärtused erinevate koormusnurkade all. Koormusnurga δ_k definitsiooni on seletatud alapeatükis 0. Täpsustus – siin on tegu rootori mehaanilise nurgaga. Momendi T mõõtmiseks kasutatakse tarkvara *FEMM* ning seda mõõdetakse 1° sammuga. Tulenevalt mähise ebasümmeetriast vaadeldakse koormusnurka nii positiivsel kui ka negatiivsel poolel (võrrelduna nende keskele jääva nullpunktiga). Mähise voolu amplituudväärtuseks valitakse $I_m = 5$ A. Tulemusena saadud kõverad esialgse rootori, juhtumi 1 ja juhtumi 2 korral on toodud joonisel 6.7. Jooniselt on näha, et esialgse rootori korral on momendi pulsatsioon kõige madalam. Juhtumi 1 ja juhtumi 2 korral on näha momendi suurem pulsatsioon.



Joonis 6.7 Momendi T sõltuvus koormusnurgast esialgse rootori, juhtum 1 ja juhtum 2 korral

Pulsatsiooni täpsemaks hindamiseks kasutatakse Fourier' teisendust, mis teostatakse *MATLAB*-i abil. Tulemusena saadud sagedusspekter on toodud joonisel 6.8. Põhiharmoonilise sagedus on 100 Hz, kuna momendikõvera puhul on sagedus kaks korda kõrgem voolu sagedusest (50 Hz). Põhiharmooniliste amplituudväärtused on järgmised: esialgne rootor – 24,8 Nm; juhtum 1 – 25,1 Nm; juhtum 2 – 26,2 Nm. Vaadeldes kõrgemaid harmoonilisi on kõige suurem mõju 2. ja 8. harmoonilisel. Märkatav mõju on ka suhteliselt kõrget järku 16., 17. ja 18. harmoonilisel, aga seda ainult optimeeritud juhtumi 1 ja juhtumi 2 korral.



Joonis 6.8 Momendikõverate sagedusspekter esialgse rootori, juhtum 1 ja juhtum 2 korral

Pulsatsioonimomenti mõjutab kõige enam tõkete välimise otsa asetus ja selle geomeetria [41, 9, 42]. Lisaks tõkete optimeerimisele on pulsatsioonimomendi vähendamiseks mitmeid võimalusi – rootori üle pikkuse mingi nurga võrra pööramine [43], erikujulise rootori kasutamine [44], püsimagnetite kasutamine (lisaks nende asetuse optimeerimine) [45], toitevoolu optimeerimine [2].

KOKKUVÕTE

Töö eesmärgiks oli töötada välja optimeerimismeetod sünkroon-reluktantsmootori efektiivsuse tõstmiseks. Kõrgem efektiivsus võimaldab kokku hoida ressursse ning tarbitavat elektrienergiat. Sünkroon-reluktantsmootorit võrreldakse tööstuses eelkõige asünkroonmootoriga tulenevalt nende sarnastest omadustest. Sünkroon-reluktantsmootori eelisteks on kõrgem saavutatav efektiivsus, ülekoormatavus, moment ruumala kohta ning lihtsam rootori konstruktsioon. Peamiseks puuduseks on suhteliselt kõrge võimsustegur. Staatori ehitus on sama nagu sünkroon- ja asünkroonmootoril. Õhupilu magnetvootiheduse siinuselise lainekuju saavutamiseks kasutatakse mitmekihilist mähist. Rootori ehituse osas on tähtis võimalikult kõrge mähise induktiivsuste erinevus d- ja q-telje asenditel. Rootor koostatakse võlliga risti asetsevatest teraslehtedest, mille õhutühimikud toimivad magnetvoo tõkete ja suunajatena.

Modelleerimise ja arvutuste lihtsustamiseks vaadeldakse suurusi dq-ristteljestikus. Sünkroon-reluktantsmootori projekteerimisel on üheks tähtsaimaks asjaoluks d- ja q-telje induktiivsuste L_d , L_q piisava (maksimaalse) erinevuse saavutamine. Induktiivsuste vahest $L_d - L_q$ sõltub otseselt mootori keskmine moment ning suhtest L_d/L_q sõltub mootori võimsustegur. Antud töö peamiseks eesmärgiks oli töötada välja automaatne optimeerimismeetod sünkroon-reluktantsmootori rootori magnetvoo tõkete sellise asetuse määramiseks, millele vastab maksimaalne suhe L_d/L_q .

Optimeerimismeetod põhineb mootori lihtsustatud magnetahelal. Ahela struktuur on erinev d- ja q-telje korral. Mõlemal juhul sõltub struktuur rootori magnetvoo tõkete asetusest. Ahela automaatse koostamise põhimõtte selgitus mistahes tõkete asetuse korral on toodud voodiagrammi abil lisa 1. Kuna d- ja q-telje ahelad koostatakse sõlmede kaupa, kasutatakse lahendamiseks sõlmpingemeetodit. Lekkevoogu arvestatakse ainult õhu takistuste juures eksperimentaalsel teel saadud koefitsiendiga ($\lambda = 1,12$). Tulenevalt terase magnetilise takistuse mittelineaarsusest kasutatakse sobiva lahendi leidmiseks iteratiivset protsessi, mille tulemusena leitakse igale takistusele sobilik suhteline magnetiline läbitavus μ_r . Õhupilu aheldusvoog leitakse staatori hammaste aheldusvoogude summana. Seejuures sõltub tulemus staatori mähkimisviisist, mida arvestatakse lihtsustatud mähkimisteguriga ($k_w = 0,917$). Lahendamise lõpptulemusena saadakse induktiivsuste väärtused L_d ja L_q . Kogu protsessi automatiseerimiseks on lisa 2 toodud *MATLAB* programmikood. Optimeerimisprobleemi lahendamine sõltub suuresti selle eripäradest, piirangutest ja sisendparameetritest. Töös toodud meetod eeldab, et eelnevalt on fikseeritud mootori peamised mõõtmed ning staatori ja selle mähise ehitus. Rootori osas on eelnevalt määratud magnetvoo tõkete arv ja nende mõõtmete piirid.

Antud töös on kirjeldatud optimeerimisprotsessi etteantud 10,3 kW mootori näitel. Probleemi lahendamiseks kasutati tarkavara *MathWorks MATLAB* lahendajat *genetic algorithm*. *Genetic algorithm* lahendaja liigub järk-järgult globaalse ekstreemumi poole kasutades eelnevatel sammudel leitud punkte. Optimeerimisprobleemi muutujateks on magnetvoo tõkete mõõtmed. Maksimaalseks magnetvootiheduseks mootoris (magnetilise küllastuse tase) valiti ligikaudu 1,6 T. Optimeerimisprotsess viidi läbi kahe juhtumi jaoks. Juhtumi 1 korral fikseeriti rootori sildade laiused esialgse rootori järgi. Juhtumi 2 korral kasutati parendatud sildade laiusi.

Magnetahela põhjal saadud tulemusi võrreldi tarkavara *FEMM* abil saadud tulemustega. Esialgse rootoriga saadi suhte väärtuseks $L_d/L_q = 4,14$ (*FEMM* $L_d/L_q = 3,87$). Optimeerimise tulemusel saadi juhtumi 1 suhte väärtuseks $L_d/L_q = 4,82$ (*FEMM* $L_d/L_q = 4,10$). Juhtum 2 korral saadi suhte väärtuseks $L_d/L_q = 5,83$ (*FEMM* $L_d/L_q = 5,20$). *FEMM* mudel põhineb lõplike elementide meetodil, mis on oluliselt täpsem. Suhteliselt suur erinevus mudelite vahel tuleneb sellest, et optimeerimisega leitakse ühtlasi magnetahelal põhineva mudeli suurim viga. Võrreldes üksnes *FEMM* mudeli väärtusi on optimeeritud suhe L_d/L_q juhtumi 1 korral ligikaudu 6,1% kõrgem ja juhtumi 2 korral ligikaudu 34,4% kõrgem esialgsest väärtusest. Tulemuste *FEMM* mudeli põhjal on näha, et magnetvootihedus on rootoris jaotunud ühtlasemalt ja rootori materjal on kasutatud efektiivsemalt. Mudelite valideerimiseks mõõdeti füüsilise 10,3 kW mootori induktiivsused katseliselt staatilises olekus. Erinevus katseliselt mõõdetud ja *FEMM* mudeli induktiivsuste vahel on d-telje korral 8,0% ja q-telje korral 2,9%.

Üheks negatiivseks asjaoluks suhte L_d/L_q maksimeerimisel on pulsatsioonimomendi suurenemine, mille hindamiseks leiti momendikõverad *FEMM* mudeli põhjal. Momendikõverate sagedusspektrilt on näha, et mõlema optimeeritud juhtumi korral on suurenenud kõrgemate harmooniliste osakaal. Antud töö edasiarenduseks on võimalik kasutada mitme-eesmärgilist optimeerimist, millega maksimeeritakse suhet L_d/L_q ja minimeeritakse pulsatsioonimomenti. Lisaks on võimalik parendada mootori magnetahelat täpsustades magnetiliste takistuste geometria määramist ning arvestades täpsemini lekkevoogu. Mehaaniliselt poolelt on edasiarenduseks võimalik leida rootori sildade optimaalsed (minimaalsed) laiused, milleks on vajalik mehaaniline tugevusarvutus sõltuvalt rootori maksimaalsest pöörlemiskiirusest.

SUMMARY

The goal of this work was to design optimization method to increase the efficiency of synchronous reluctance machine. With higher efficiency resources can be saved and energy usage reduced. In industry synchronous reluctance machine is mainly compared to induction machine because of their similarities. Advantages of synchronous reluctance machine are higher achievable efficiency, overloading, torque density and simpler construction of the rotor. Main disadvantage is relatively high power factor. Stator construction is the same as synchronous and induction machine. Multilayer winding is used to achieve sine waveform for magnetic density in air gap. In rotor it is important to achieve high inductance difference in d- and q-axis positions. Rotor is made of metal sheets which have hollow areas acting as magnetic flux barriers and guides.

Values are given in dq-frame to simplify modelling and calculations. One of the most important aspects when designing synchronous reluctance machine is to achieve sufficient (maximal) difference between d- and q-axis inductances L_d, L_q . Motor average torque is directly dependent on difference $L_d - L_q$ and power factor is dependent on ratio L_d/L_q . The main goal of given work was to design automated optimization method for determining the placement of flux barriers for synchronous reluctance machine in a way to maximize the ratio L_d/L_q .

Optimization method is based on simplified magnetic circuit of motor. Circuit structure is different for d- and q-axis. In both cases the circuit structure depends on the placement of flux barriers. The principle for automated circuit creation with any kind of barrier placement is given as flowchart in appendix 1. Because d- and q-axis circuits are created node by node the node voltage method is used. Leakage flux is taken into account only with air reluctances using experimentally found coefficient ($\lambda = 1,12$). Since steel has nonlinear reluctance the iterative process is used to find suitable solution. As a result relative magnetic permeability μ_r is found for each reluctance. The linkage flux in air gap is found as a sum of stator teeth linkage fluxes. This result depends also on stator winding layout, which is taken into account as a simplified winding factor ($k_w = 0,917$). Inductance values L_d and L_q are acquired as the end result. To automate this whole process *MATLAB* program code is given in appendix 2. Solving the optimization problem depends on its properties, limitations and input parameters. The method given in this work assumes that main dimensions of the motor and also stator and its windings are previously defined and fixed. In rotor the number of flux barriers and their dimensions limits have to be previously defined.

In given work the optimization process is described based on given 10,3 kW motor. Software *MathWorks MATLAB* solver *genetic algorithm* was used to solve the optimization problem. *Genetic algorithm* solver moves step by step towards the global optima while using points found on previous steps. Optimization problem variables are the dimensions of flux barriers. Maximum magnetic flux density (magnetic saturation level) in motor was chosen to be approximately 1,6 T. Optimization process was carried out for two cases. For case 1 the widths of rotor bridges were fixed according to initial rotor. For case 2 the widths of rotor bridges were improved.

Results based on magnetic circuit were compared to results gotten with software *FEMM*. With initial rotor design the ratio value is $L_d/L_q = 4,14$ (*FEMM* $L_d/L_q = 3,87$). Looking at optimized results for case 1 the ratio value is $L_d/L_q = 4,82$ (*FEMM* $L_d/L_q = 4,10$). For case 2 the ratio value is $L_d/L_q = 5,83$ (*FEMM* $L_d/L_q = 5,20$). *FEMM* is based on finite element method which is more precise. There is relatively large difference between two models because with optimization also the biggest error for magnetic circuit is found. When comparing only results gotten with *FEMM* model the optimized ratio L_d/L_q with case 1 is about 6,1% higher and with case 2 about 34,4% higher when compared to initial value. From *FEMM* model it is seen for both cases that in rotor the magnetic flux density is more evenly distributed and material is used more efficiently. For validation of the models inductances of physical 10,3 kW motor were measured experimentally in static state. Differences between experimentally measured and *FEMM* model inductances are 8,0% for d-axis and 2,9% for q-axis.

When maximising the ratio L_d/L_q one negative aspect is the increase of torque ripple. For torque ripple evaluation the torque characteristics were found using *FEMM* model. From frequency spectrum of torque characteristics it is seen that for both optimized cases the proportion of higher frequencies has increased. For future development it is possible to use multiobjective optimization to maximize the ratio L_d/L_q and minimize torque ripple. Furthermore it is possible to improve magnetic circuit by making geometry of reluctances and flux leakage consideration more accurate. For future development in mechanical side it is possible to find rotor bridges optimal (minimal) widths which requires mechanical stress calculations dependent on rotor maximal rotation speed.

KASUTATUD KIRJANDUS

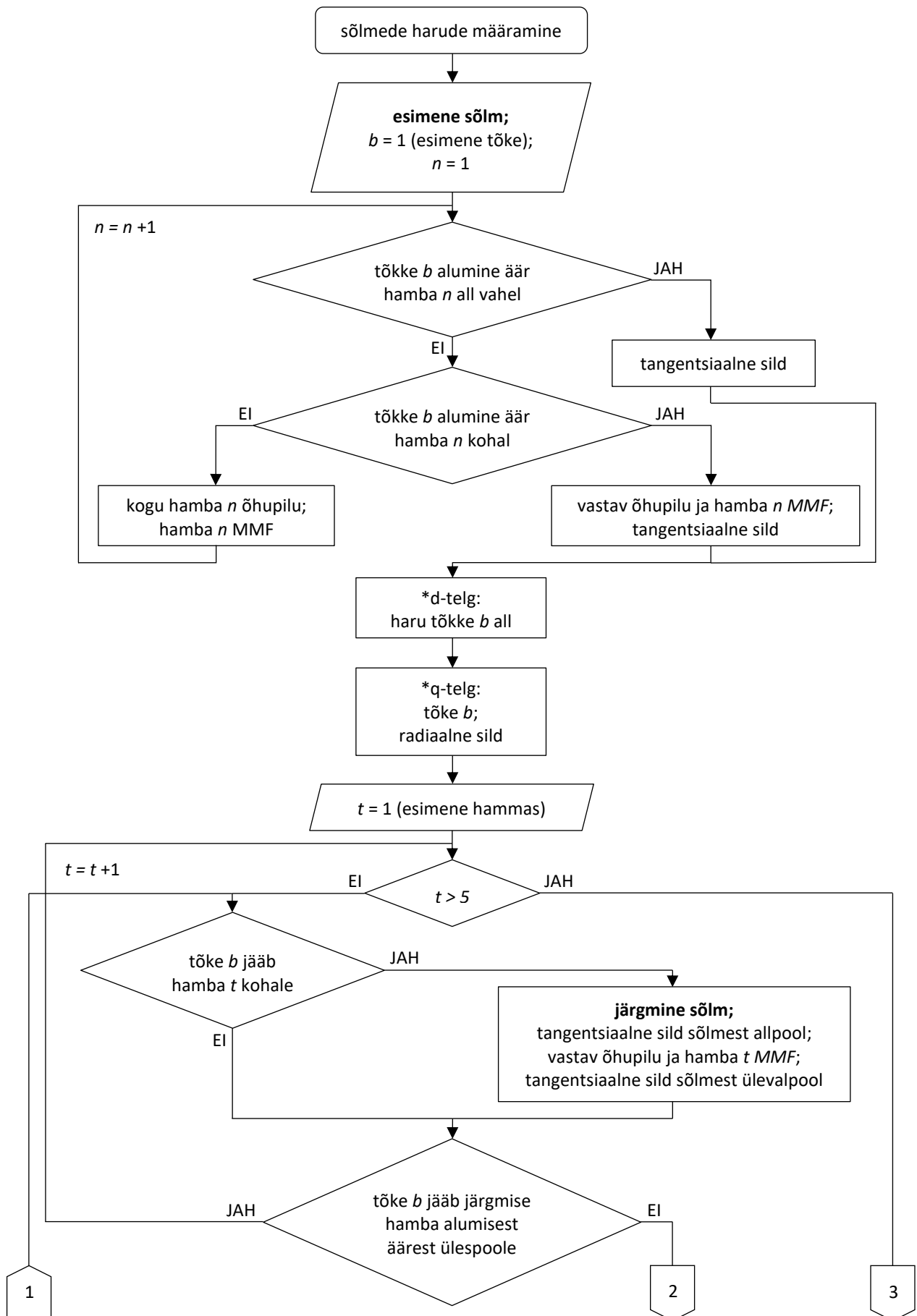
- [1] C. R. D., J. F. F. ja S. L., „Current Trends in Energy Efficient Electrical Machines,” *Journal of Electrical and Electronics Engineering*, kd. 10, lk 13-18, 2017.
- [2] H. Wu, D. Depernet ja V. Lanfranchi, „Analysis of torque ripple reduction in a segmented-rotor synchronous reluctance machine by optimal currents,” *Mathematics and Computers in Simulation*, kd. 158, lk 130-147, 2019.
- [3] S. Jia, P. Zhang, D. Liang, M. Dai ja J. Liu, „Design of IE4 Level Synchronous Reluctance Machines with Different Number of Poles,” *22nd ICEMS*, Harbin, China, 2019.
- [4] „IE4 synchronous reluctance motors,” ABB Motors and Generators, 2016. [Vörgumaterjal].
Saadaval:
https://library.e.abb.com/public/6d4f5034141d4e0eb9f30a1abc6445fc/IE4_SynRM_08-2016_LOW.pdf [Kasutatud 13.12.19].
- [5] C. L. Torres, "Analysis and implementation of a methodology for optimal PMA-SynRM design taking into account performances and reliability," Doctoral thesis, Polytechnic University of Catalonia.
- [6] A. J. P. Ortega, "Design and Comparison of Induction Motor and Synchronous Reluctance Motor for Variable Speed Applications: Design Aided by Differential Evolution and Finite Element Analysis", Master Thesis, The Ohio State University, 2013.
- [7] R. Moghaddam, "Synchronous Reluctance Machine (SynRM) in Variable Speed Drives (VSD) Applications," Doctoral Thesis, Stockholm, Sweden: KTH School of Electrical Engineering, 2011.
- [8] S. Sathyan, "Synchronous Reluctance Motor for Household Applications," Masters thesis, Espoo: Aalto University, 2013.
- [9] C. Babetto, G. Bacco ja N. Bianchi, „Synchronous Reluctance Machine Optimization for High-Speed Applications,” *IEEE Transactions on Energy Conversion*, kd. 33, nr 3, lk 1266-1273, 2018.
- [10] M. H. Mohammadi, T. Rahman ja D. Lowther, „Restricting the design space of multiple-barrier rotors of synchronous reluctance machines,” *COMPEL*, kd. 36, nr 5, lk 1338-1350, 2017.
- [11] D. A. ja M. C., „Simplified model of synchronous reluctance machine with optimized flux barriers,” *Electr Eng (2017)*, nr 99, lk 1207–1216, 2017.

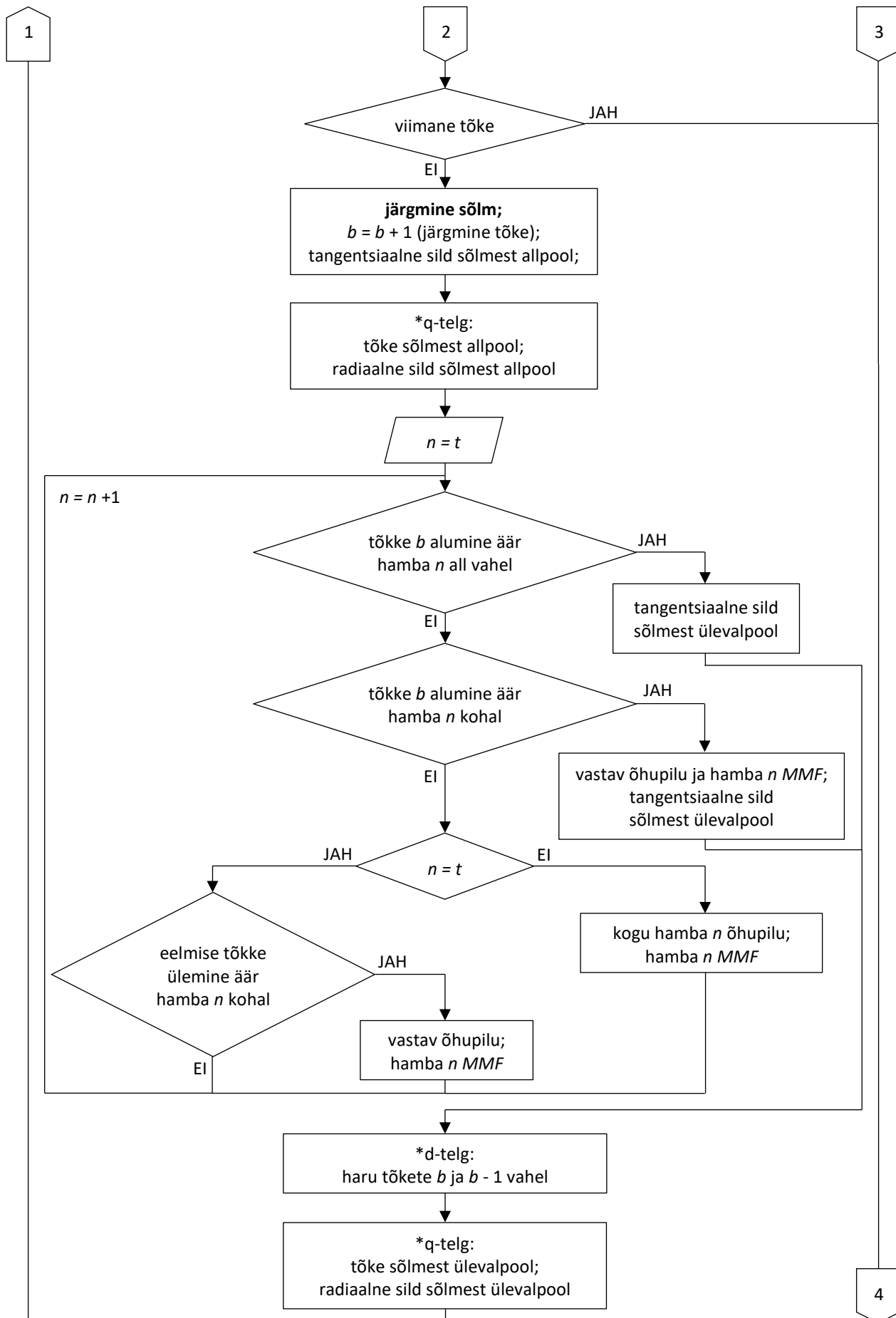
- [12] S. Yammine, C. Henaux, M. Fadel, S. Desharnais ja L. Calegari, „Synchronous Reluctance Machine Flux Barrier Design based on the Flux Line Patterns in a Solid Rotor,“ *2014 International Conference on Electrical Machines (ICEM)*, Berlin, Germany, 2014.
- [13] G. Bacco ja N. Bianchi, „Design Criteria of Flux-Barriers in Synchronous Reluctance Machines,“ *IEEE Transactions on Industry Applications*, kd. 55, nr 3, lk 2490-2498, 2019.
- [14] I. M., S. P. ja R. E. E., „Simple Design Approach for Low Torque Ripple and High Output Torque Synchronous Reluctance Motors,“ *Energies*, kd. 9, 2016.
- [15] A. Hughes, *Electric Motors and Drives, Fundamentals, Types and Applications*, Third edition toim., Elsevier Ltd., 2006.
- [16] B. Bose, *Modern Power Electronics and AC Drives*, Knoxville. Prentice Hall PTR: The University of Tennessee, 2002.
- [17] V. Vodovozov, D. Vinnikov ja R. Jansikene, *Elektriamite elektroonsed süsteemid*, Tallinn: TTÜ elektriamite ja jõuelektronika instituut, 2008.
- [18] C. W. Brice, E. Santi, X. Kang ja L. U. Gökdere, *Reference frames for simulation of electric motors and drives*, Columbia: University of South Carolina, 2002.
- [19] „Coordinate Transform in Motor Control,“ Cypress Semiconductor Corp, [Võrgumaterjal]. Saadaval: <http://www.cypress.com/file/222111/download> [Kasutatud 29.11.2018].
- [20] I. C. Proimadis, D. V. Spyropoulos ja E. D. Mitronikas, „An Alternative for All-Electric Ships Applications: The Synchronous Reluctance Motor,“ *Advances in Power Electronics*, kd. 2013, nr ID 862734, 2013.
- [21] T. Mohanarajah, J. Rizk, M. Nagrial ja A. Hellany, „Design of Synchronous Reluctance Motors with Improved Power Factor,“ *2017 11th IEEE CPE-POWERENG*, Cadiz, Spain, 2017.
- [22] „Elektromehaaniline energiamuundus, sünkroonmasin,“ Aine SKK0070 õppeinfo, [Võrgumaterjal]. Saadaval: http://www.ene.ttu.ee/elektriamid/oppeinfo/materjal/SKK0070/032Elektromehaaniline_energiamuundus.pdf. [Kasutatud 29.11.2018].
- [23] D. A. Staton, T. J. E. Miller ja S. E. Wood, „Maximising the saliency ratio of the synchronous reluctance motor,“ *IEE PROCEEDINGS-B*, kd. 140, nr 4, lk 249-259, 1993.
- [24] U. A. Bakshi ja V. U. Bakshi, *Basics for electrical engineering*, Pune, India: Technical Publications, 2008.

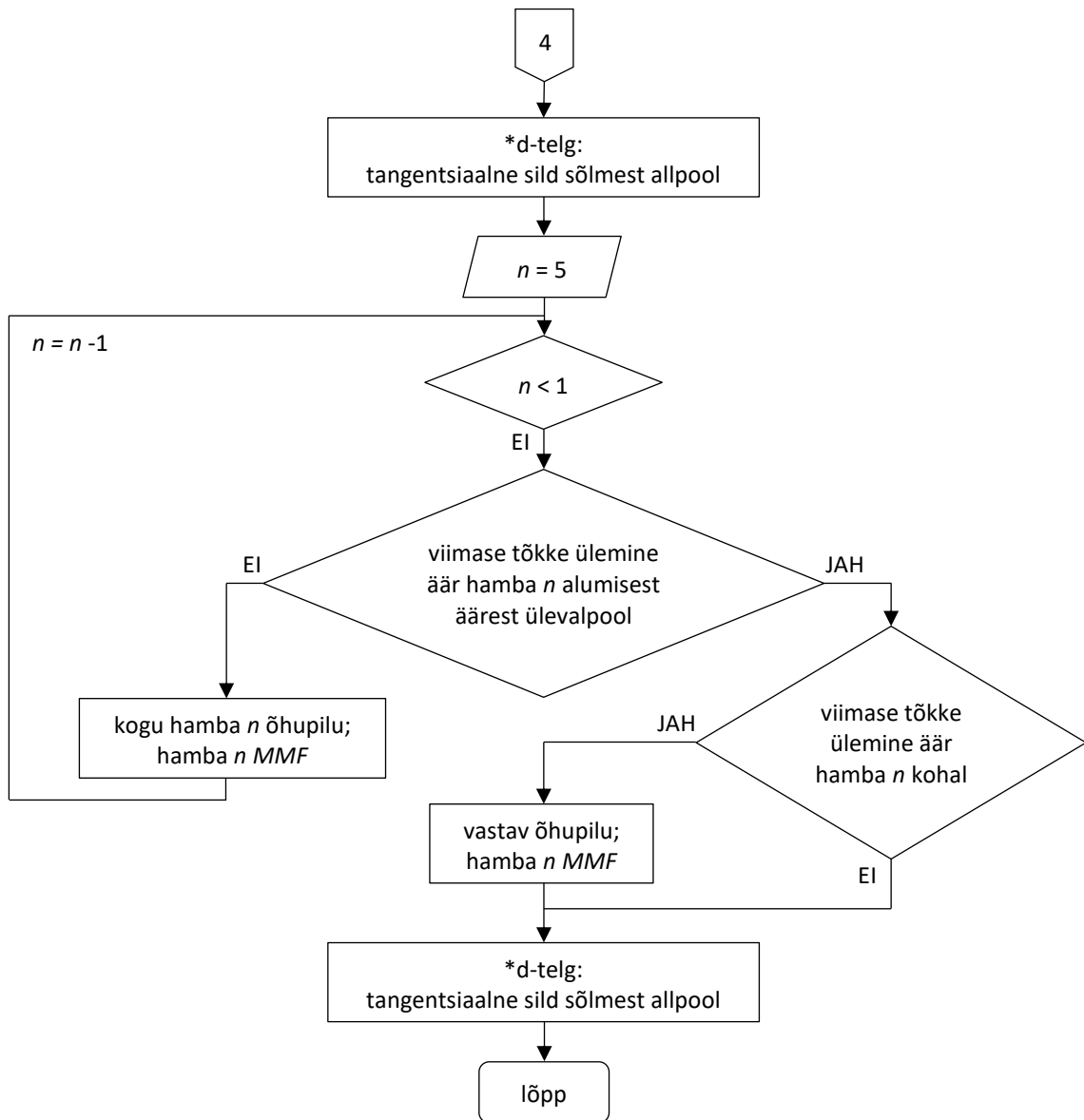
- [25] R. Vartanian, H. A. Toliyat, B. Akin ja R. Poley, „Power Factor Improvement of Synchronous Reluctance Motors (SynRM) Using Permanent Magnets for Drive Size Reduction,“ *2012 Twenty-Seventh Annual IEEE APEC*, Orlando, FL, USA, 2012.
- [26] N. Bianchi., *Synchronous Reluctance and PM Assisted Reluctance Motors*, University of Padova, 2016.
- [27] K. Janson, *Elektrimasinad: loengukonspekt*, Tallinn: Tallinna Tehnikaülikool, 2012.
- [28] B. Pinnock ja T. Verkest, "Magnetic Circuit Model for Synchronous Reluctance Motors," Master thesis, Ghent University, 2017.
- [29] H. Grop, "Investigation of AC Electrical Machine Stators with Fractional Conductor Windings", Stockholm, Sweden: KTH School of Electrical Engineering, 2010.
- [30] „Solver-Based Optimization Problem Setup,“ The MathWorks, Inc., [Võrgumaterjal]. Saadaval: <https://uk.mathworks.com/help/optim/optimization-problem-setup-solver-based.html> [Kasutatud 08.11.19].
- [31] M. S. Yammine, "Contribution to the Synchronous Reluctance Machine Performance Improvement," Doctoral Thesis, Universite de Toulouse, 2015.
- [32] S. D. Sudho, J. Cale, B. Cassimere ja M. Swinney, „Genetic Algorithm Based Design of a Permanent Magnet Synchronous Machine,“ *IEEE International Electric Machines and Drives Conference*, lk 1011-1019, 2005.
- [33] G. Pellegrino, F. Cupertino ja C. Gerada, „Automatic Design of Synchronous Reluctance Motors focusing on Barrier Shape Optimization,“ *IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS*, kd. 51, nr 2, lk 1465-1474, 2015.
- [34] „How GlobalSearch and MultiStart Work,“ The MathWorks, Inc., [Võrgumaterjal]. Saadaval: <https://uk.mathworks.com/help/gads/how-globalsearch-and-multistart-work.html> [Kasutatud 08.11.19].
- [35] H. T., S. R., R. M., K. A. ja H. H. G., „Nonlinear Reluctance Network Method for Synchronous Reluctance Machine Analysis,“ *2015 18th ICEMS*, Pattaya City, 2015.
- [36] S. H. Hwang, J. M. Kimy, H. V. Khang ja J. W. Ahn, „Parameter Identification of a Synchronous Reluctance Motor by using a Synchronous PI Current Regulator at a Standstill,“ *Journal of Power Electronics*, kd. 10, nr 5, lk 491-497, 2010.
- [37] D. Fernando ja V. Garcia, "Self-commissioning and testing of synchronous reluctance motor drives," Masters thesis, Universidad de Oviedo, 2017.

- [38] D. Žarko, D. Ban ja R. Klaric, „Finite Element Approach to Calculation of Parameters of an Interior Permanent Magnet Motor,“ *AUTOMATIKA*, kd. 46, nr 3-4, lk 113-122, 2005.
- [39] „Genetic Algorithm Options,“ The MathWorks, Inc., [Võrgumaterjal]. Saadaval: <https://uk.mathworks.com/help/gads/genetic-algorithm-options.html> [Kasutatud 15.11.19].
- [40] V. Bobek, „PMSM Electrical Parameters Measurement,“ Freescale Semiconductor, Inc., 2013. [Võrgumaterjal]. Saadaval: <https://www.nxp.com/docs/en/application-note/AN4680.pdf> [Kasutatud 29.11.19].
- [41] R. Johanson, "Püsimagnetitega sünkroonreluktantsmasina momendikarakteristikute modelleerimine," magistritöö, Tallinn, 2015.
- [42] M. Sanada, K. Hiramoto ja S. Morimoto, „Torque ripple improvement for synchronous reluctance motor using asymmetric flux barrier arrangement,“ *IEEE Industry Applications Conference IAS Annual Meeting*, Salt Lake City, 2003.
- [43] A. Fratta, G. P. Trogia ja A. Vagati, „Evaluation of torque ripple in high performance synchronous reluctance machines,“ *IEEE Industry Applications Society Annual Meeting*, New York, 1993.
- [44] M. Muteba, B. Twala ja D. V. Nicolae, „Torque Ripple Minimization in Synchronous Reluctance Motor Using a Sinusoidal Rotor Lamination Shape,“ *2016 XXII International Conference on Electrical Machines (ICEM)*, Lausanne, 2016.
- [45] M. H. Hwang, H. S. Lee ja H. R. Cha, „Analysis of Torque Ripple and Cogging Torque Reduction in Electric Vehicle Traction Platform Applying Rotor Notched Design,“ *Energies*, kd. 11, nr 3053, 2018.

Lisa 1 Lihtsustatud voodiagramm ahela sõlmedele vastavate harude määramiseks







Lisa 2 MATLAB programmikood induktiivsuste L_d , L_q leidmiseks

L2.1 Programmikoodi põhiosa

```

%% stator winding information
PH=3; %number of phases
NT=23; %23 turns in one layer
I=5;
%shift -20 mech. degrees (-40 el. degrees)
IA=I*sin((pi*(-40))/180);
IB=I*sin(0+2/3*pi+(pi*(-40))/180);
IC=I*sin(0+2/3*2*pi+(pi*(-40))/180);
IX=-IA;
IY=-IB;
IZ=-IC;
WP=[ %winding placement matrix [tooth1 tooth2 ...]
-3 -2 0 2 3
 3 3 3 3 2
-2 -3 -3 -3 -3];
%MMF matrix - [F1 F2 F3 F4 F5]
F(1,:)=(WP(1,:)*IA*NT);
F(2,:)=(WP(2,:)*IB*NT);
F(3,:)=(WP(3,:)*IC*NT);
F=sum(F);
kw=0.917; %winding factor
%% machine dimensions
EL=156; %rotor effective length
Dsi=136; %inner diameter of stator
Dso=219; %outer diameter of stator
Dri=45; %inner diameter of rotor
Dro=135.2; %outer diameter of rotor
ag=(Dsi-Dro)/2; %air gap
MTA=[ %matrix tooth angles - [tooth t lower edge, tooth t upper edge]
0 4.1574
5.8426 14.1574
15.8426 24.1574
25.8426 34.1574
35.8426 44.1574];
%% barrier dimensions
nb=4; %number of barriers (per pole)
MBI=[ %matrix barrier information - [B1 B2 B3 B4 B5 B6 B7 B8]
7 2.5 21.886 1 4 3.4495 3.8759 67.5
16 2.5 18.1581 1 4 3.5137 11.6970 67.5
25 2.5 14.4302 1 4 3.6535 19.7381 67.5
34 2.5 10.7023 1 4 3.8984 28.1921 67.5];
%% barrier locations
%MPX - matrix placement x-axis - [X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14]
%MPY - matrix placement y-axis - [Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10 Y11 Y12 Y13 Y14]
for b=1:nb
MPX(b,1)=MBI(b,2)/2;
MPY(b,1)=Dri/2+MBI(b,1)-MBI(b,5)/2;
MPX(b,2)=MPX(b,1);
MPY(b,2)=Dri/2+MBI(b,1)+MBI(b,5)/2;
MPX(b,3)=MPX(b,1)+MBI(b,3);
MPY(b,3)=MPY(b,1)+MBI(b,5)/2;
MPY(b,4)=MPY(b,1);
MPX(b,4)=MPX(b,3)+(MPY(b,4)-MPY(b,3))/-tan((MBI(b,8)*pi)/180);
MPY(b,5)=MPY(b,2);
MPX(b,5)=MPX(b,3)+(MPY(b,5)-MPY(b,3))/-tan((MBI(b,8)*pi)/180);
MPX(b,6)=cos((45+MBI(b,7)-MBI(b,6)/2)*pi/180)*(Dro/2-MBI(b,4));
MPY(b,6)=sin((45+MBI(b,7)-MBI(b,6)/2)*pi/180)*(Dro/2-MBI(b,4));
MPX(b,7)=cos((45+MBI(b,7)+MBI(b,6)/2)*pi/180)*(Dro/2-MBI(b,4));
MPY(b,7)=sin((45+MBI(b,7)+MBI(b,6)/2)*pi/180)*(Dro/2-MBI(b,4));
MPX(b,8)=MBI(b,2)/2;
MPY(b,8)=MPY(b,3);
MPX(b,9)=cos((45+MBI(b,7))*pi/180)*(Dro/2-MBI(b,4));
MPY(b,9)=sin((45+MBI(b,7))*pi/180)*(Dro/2-MBI(b,4));
MPX(b,10)=(MPX(b,9)-MPX(b,3))/2+MPX(b,3);
MPY(b,10)=(MPY(b,9)-MPY(b,3))/2+MPY(b,3);
k=- (MPX(b,9)-MPX(b,3))/(MPY(b,9)-MPY(b,3)); %slope between 11, 12
MPX(b,11)=(-MPY(b,10)*MPX(b,4)+k*MPX(b,10)*MPX(b,4)+MPY(b,6)*MPX(b,4)+MPY(b,10)*MPX(b,6)-
k*MPX(b,10)*MPX(b,6)-MPX(b,6)*MPY(b,4))/(k*MPX(b,4)-k*MPX(b,6)-MPY(b,4)+MPY(b,6));
MPY(b,11)=MPY(b,10)+k*MPX(b,11)-k*MPX(b,10);
MPX(b,12)=(-MPY(b,10)*MPX(b,5)+k*MPX(b,10)*MPX(b,5)+MPY(b,7)*MPX(b,5)+MPY(b,10)*MPX(b,7)-
k*MPX(b,10)*MPX(b,7)-MPX(b,7)*MPY(b,5))/(k*MPX(b,5)-k*MPX(b,7)-MPY(b,5)+MPY(b,7));
MPY(b,12)=MPY(b,10)+k*MPX(b,12)-k*MPX(b,10);

```

```

MPX(b,13)=cos((45+MBI(b,7)-MBI(b,6)/2)*pi/180)*(Dro/2);
MPY(b,13)=sin((45+MBI(b,7)-MBI(b,6)/2)*pi/180)*(Dro/2);
MPX(b,14)=cos((45+MBI(b,7)+MBI(b,6)/2)*pi/180)*(Dro/2);
MPY(b,14)=sin((45+MBI(b,7)+MBI(b,6)/2)*pi/180)*(Dro/2);
end
%MBA - matrix barrier angles - [barrier b end lower edge, middle, upper edge]
for b=1:nb
    MBA(b,1)=MBI(b,7)-MBI(b,6)/2;
    MBA(b,2)=MBI(b,7);
    MBA(b,3)=MBI(b,7)+MBI(b,6)/2;
end
%% D_AXIS
%% declaring variables
i=0; %node counter
j=0; %branch (flux) counter
b=0; %barrier counter
rs=0; %steel reluctance counter
ra=0; %air reluctance counter
%info - [reluctance,length,width,B,B',mu,i,j,permeance,next node,MMF,1-connected to last
node,1-multiple reluctances,1-radial bridge,width2]
M3=zeros(1,15); %steel reluctance info matrix
M4=zeros(1,15); %air reluctance info matrix
S=zeros(1,5); %[tooth1 tooth2 ...] 0-node at tooth end; 1-no node at tooth end
%% first node
i=i+1;
b=b+1;
for n=1:5 %finding barrier
    if MBA(b,1)<MTA(n,1) %if barrier lower edge is on lower slot
        j=j+1;
        rs=rs+1;
        if MBA(b,3)<MTA(n,1) %if barrier upper edge is on lower slot
            M3(rs,2)=(MBA(b,3)-MBA(b,1))*(Dro*pi/360);
        elseif MBA(b,3)<MTA(n,2) %if barrier upper edge is on tooth
            M3(rs,2)=(MBA(b,3)-MTA(n,1))/4+MTA(n,1)-MBA(b,1)*(Dro*pi/360);
        else %if barrier upper edge is above tooth
            M3(rs,2)=(MTA(n,2)-MTA(n,1))/4+MTA(n,1)-MBA(b,1)*(Dro*pi/360);
        end
        break
    end
    if MBA(b,1)<MTA(n,2) %if barrier lower edge is on tooth
        %airgap reluctance and MMF
        j=j+1;
        ra=ra+1;
        M4(ra,2)=ag;
        M4(ra,3)=(MBA(b,1)-MTA(n,1))*(Dsi*pi/360);
        M4(ra,7)=i;
        M4(ra,8)=j;
        M4(ra,11)=n;
        %tangential bridge above node
        j=j+1;
        rs=rs+1;
        if MBA(b,3)<MTA(n,2) %if barrier upper edge is on tooth
            M3(rs,2)=(MBA(b,2)-MBA(b,1))/2*(Dro*pi/360);
        else %if barrier upper edge is above tooth
            M3(rs,2)=(MTA(n,2)-MBA(b,1))/4*(Dro*pi/360);
        end
        break
    end
    %airgap reluctance and MMF
    j=j+1;
    ra=ra+1;
    M4(ra,2)=ag;
    M4(ra,3)=(MTA(n,2)-MTA(n,1))*(Dsi*pi/360);
    M4(ra,7)=i;
    M4(ra,8)=j;
    M4(ra,11)=n;
    M4(ra,13)=1;
    %tooth reluctances
    if n==1
        [M3,rs]=tooth_first(M3,rs,i,j,n);
    else
        [M3,rs]=tooth(M3,rs,i,j,n);
    end
    S(n)=1;
end
M3(rs,3)=MBI(b,4);
M3(rs,7)=i;

```

```

M3(rs,8)=j;
%reluctance of area below barrier
j=j+1;
rs=rs+1;
[M3(rs,2),M3(rs,3),~,~,~]=barrier_lower_first(MPX,MPY);
M3(rs,7)=i;
M3(rs,8)=j;
M3(rs,12)=1;
M3(rs,13)=1;
rs=rs+1;
[~,~,M3(rs,2),M3(rs,3),M3(rs,15)]=barrier_lower_first(MPX,MPY);
M3(rs,7)=i;
M3(rs,8)=j;
M3(rs,12)=1;
M3(rs,13)=1;
%% intermediate nodes
for t=1:5
    while 1 %repeat if there are more than one barriers on tooth or on above slot
        if MBA(b,3)>MTA(t,1) && MBA(b,1)<MTA(t,2) %if barrier is on tooth
            %node on barrier end
            %tangnetial bridge below node
            i=i+1;
            rs=rs+1;
            if t==1
                M3(rs,:)=M3(rs-3,:);
                M3(rs,10)=M3(rs,7);
            else
                if MBA(b,1)>MTA(t-1,2) %if barrier lower edge is on tooth or on lower
slot
                    M3(rs,:)=M3(rs-3,:);
                    M3(rs,10)=M3(rs,7);
                else %if barrier is on lower tooth
                    M3(rs,:)=M3(rs-1,:);
                    M3(rs,10)=0;
                end
            end
            M3(rs,7)=i;
            %airgap reluctance and MMF
            j=j+1;
            ra=ra+1;
            M4(ra,2)=ag;
            if MBA(b,1)<MTA(t,1) && MBA(b,3)<MTA(t,2) %if barrier lower edge is under
tooth and barrier upper edge is on tooth
                M4(ra,3)=(MBA(b,3)-MTA(t,1))*(Dsi*pi/360);
            elseif MBA(b,1)<MTA(t,1) && MBA(b,3)>MTA(t,2) %if barrier lower edge is
under tooth and barrier upper edge is above tooth
                M4(ra,3)=(MTA(t,2)-MTA(t,1))*(Dsi*pi/360);
            elseif MBA(b,1)>MTA(t,1) && MBA(b,3)<MTA(t,2) %if barrier lower and upper
edge is on tooth
                M4(ra,3)=(MBA(b,3)-MBA(b,1))*(Dsi*pi/360);
            else %if barrier lower edge is on tooth and barrier upper edge is above
tooth
                M4(ra,3)=(MTA(t,2)-MBA(b,1))*(Dsi*pi/360);
            end
            M4(ra,7)=i;
            M4(ra,8)=j;
            M4(ra,11)=t;
            %tangential bridge above node
            j=j+1;
            rs=rs+1;
            if t~=1 %if not first tooth
                if MBA(b,1)<MTA(t-1,2) %if barrier lower edge is below lower tooth
upper edge
                    lt=M4(ra,3)/2;
                else
                    lt=M4(ra,3)/4;
                end
            else
                lt=M4(ra,3)/4;
            end
            if MBA(b,3)<MTA(t,2) || t==5 %if barrier upper edge is on tooth or last
tooth
                M3(rs,2)=lt;
            elseif MBA(b,3)<MTA(t+1,1) %if barrier upper edge is on upper slot
                M3(rs,2)=(MBA(b,3)-MTA(t,2))*(Dro*pi/360)+lt;
            elseif MBA(b,3)<MTA(t+1,2) %if barrier upper edge is on upper tooth
                M3(rs,2)=((MBA(b,3)-MTA(t+1,1))/2+MTA(t+1,1)-MTA(t,2))*(Dro*pi/360)+lt;

```

```

        else %if barrier upper edge is on above upper tooth
            M3(rs,2) = ((MTA(t+1,2) - MTA(t+1,1)) / 2 + MTA(t+1,1) -
MTA(t,2)) * (Dro*pi/360) + lt;
            end
            M3(rs,3) = MBI(b,4);
            M3(rs,7) = i;
            M3(rs,8) = j;
            M3(rs,10) = i+1;
        end

    if t~=5 %if not last tooth
        if MBA(b,3) > MTA(t+1,1) %if barrier upper edge is above upper tooth lower edge
            break %next tooth
        end
    end

    if b==nb %if last barrier
        break
    end

    b=b+1;
    %node between barriers
    i=i+1;
    %tangential bridge below node
    rs=rs+1;
    if MBA(b-1,3) > MTA(t,1) && MBA(b-1,1) < MTA(t,2) %if previous barrier is on tooth
        M3(rs,:) = M3(rs-1,:);
    else
        M3(rs,:) = M3(rs-3,:);
    end
    M3(rs,7) = i;
    M3(rs,10) = 0;
    for n=t:5 %finding barrier
        if MBA(b,1) < MTA(n,1) %if barrier lower edge is on lower slot
            %tangential bridge above node
            j=j+1;
            rs=rs+1;
            if MBA(b,3) < MTA(n,1) %if barrier upper edge is on lower slot
                M3(rs,2) = (MBA(b,3) - MBA(b,1)) * (Dro*pi/360);
            elseif MBA(b,3) < MTA(n,2) %if barrier upper edge is on tooth
                M3(rs,2) = ((MBA(b,3) - MTA(n,1)) / 4 + MTA(n,1) - MBA(b,1)) * (Dro*pi/360);
            else %if barrier upper edge is above tooth
                M3(rs,2) = ((MTA(n,2) - MTA(n,1)) / 4 + MTA(n,1) - MBA(b,1)) * (Dro*pi/360);
            end
            break
        end
        if MBA(b,1) < MTA(n,2) %if barrier lower edge is on tooth
            %airgap reluctance and MMF
            j=j+1;
            ra=ra+1;
            M4(ra,2) = ag;
            if n==t %if barrier is on tooth t
                M4(ra,3) = (MBA(b,1) - MBA(b-1,2)) * (Dsi*pi/360);
            else %if barrier lower edge is on tooth t
                M4(ra,3) = (MBA(b,1) - MTA(n,1)) * (Dsi*pi/360);
            end
            M4(ra,7) = i;
            M4(ra,8) = j;
            M4(ra,11) = n;
            %tangential bridge above node
            j=j+1;
            rs=rs+1;
            if MBA(b,3) < MTA(n,2) %if barrier upper edge is on tooth
                M3(rs,2) = ((MBA(b,2) - MBA(b,1)) / 2) * (Dro*pi/360);
            else %if barrier upper edge is above tooth
                M3(rs,2) = ((MTA(n,2) - MBA(b,1)) / 4) * (Dro*pi/360);
            end
            break
        end

    if n==t
        if MBA(b-1,3) < MTA(t,2) %if previous barrier upper edge is on tooth t
            %airgap reluctance and MMF
            j=j+1;
            ra=ra+1;
            M4(ra,2) = ag;
            M4(ra,3) = (MTA(t,2) - MBA(b-1,3)) * (Dsi*pi/360);

```

```

        M4(ra,7)=i;
        M4(ra,8)=j;
        M4(ra,11)=n;
    end
    else
        %airgap reluctance and MMF
        j=j+1;
        ra=ra+1;
        M4(ra,2)=ag;
        M4(ra,3)=(MTA(n,2)-MTA(n,1))*(Dsi*pi/360);
        M4(ra,7)=i;
        M4(ra,8)=j;
        M4(ra,11)=n;
        M4(ra,13)=1;
        %tooth reluctances
        if n==1
            [M3,rs]=tooth_first(M3,rs,i,j,n);
        else
            [M3,rs]=tooth(M3,rs,i,j,n);
        end
        S(n)=1;
    end
end
end
M3(rs,3)=MBI(b,4);
M3(rs,7)=i;
M3(rs,8)=j;
%reluctance of area below barrier
j=j+1;
rs=rs+1;
[M3(rs,2),M3(rs,3),~,~,~]=barrier_lower(MPX,MPY,b);
M3(rs,7)=i;
M3(rs,8)=j;
M3(rs,12)=1;
M3(rs,13)=1;
rs=rs+1;
[~,~,M3(rs,2),M3(rs,3),M3(rs,15)]=barrier_lower(MPX,MPY,b);
M3(rs,7)=i;
M3(rs,8)=j;
M3(rs,12)=1;
M3(rs,13)=1;
end
end
%% last node
i=i+1;
for n=5:-1:1
    if MBA(b,3)>MTA(n,1) %if barrier upper edge is above tooth lower edge
        %tangential bridge below node
        rs=rs+1;
        if MBA(b,3)>MTA(n,1) && MBA(b,1)<MTA(n,2) %if barrier is on tooth
            M3(rs,:)=M3(rs-1,:);
        else
            M3(rs,:)=M3(rs-3,:);
        end
        M3(rs,7)=i;
        M3(rs,10)=0;
        break
    end
end
%% remaining air gaps
for n=5:-1:1
    if MBA(b,3)>MTA(n,1) %if barrier upper edge is above tooth lower edge
        if MBA(b,3)<MTA(n,2)
            %airgap reluctance and MMF
            j=j+1;
            ra=ra+1;
            M4(ra,2)=ag;
            M4(ra,3)=(MTA(n,2)-MBA(b,3))*(Dsi*pi/360);
            M4(ra,7)=i;
            M4(ra,8)=j;
            M4(ra,11)=n;
        end
        break
    else
        %airgap reluctance and MMF
        j=j+1;
        ra=ra+1;
        M4(ra,2)=ag;

```

```

M4(ra,3)=(MTA(n,2)-MTA(n,1))*(Dsi*pi/360);
M4(ra,7)=i;
M4(ra,8)=j;
M4(ra,11)=n;
M4(ra,13)=1;
%tooth reluctances
[M3,rs]=tooth(M3,rs,i,j,n);
S(n)=1;
end
end
%reluctance of area above last barrier
j=j+1;
rs=rs+1;
[M3(rs,2),M3(rs,3)]=barrier_last(MPX,MPY,b,Dro);
M3(rs,7)=i;
M3(rs,8)=j;
M3(rs,12)=1;
%% nodes at teeth ends
for n=1:5
    if S(n)==0 %if there is node at tooth end
        i=i+1;
        for m=1:ra
            if M4(m,11)==n %if reluctance is according to tooth
                M4(m,10)=i; %appointing "next node" information to node in rotor
                ra=ra+1;
                M4(ra,:)=M4(m,:);
                M4(ra,7)=i;
                M4(ra,10)=0;
            end
        end
        %tooth reluctances
        j=j+1;
        if n==1
            [M3,rs]=tooth_first(M3,rs,i,j,1);
        else
            [M3,rs]=tooth(M3,rs,i,j,n);
        end
    end
end
%% nodes at stator yoke
%tooth 1 branch for node over tooth 2
i=i+1;
if S(1)==0 %if there is node at tooth 1 end
    for n=1:rs
        if M3(n,11)==1
            rs=rs+1;
            M3(rs,:)=M3(n,:);
            M3(rs,10)=M3(rs,7);
            M3(rs,7)=i;
        end
    end
else %if there is no node at tooth 1 end
    for n=1:rs
        if M3(n,11)==1
            M3(n,10)=i; %appointing "next node" information to node in rotor
            rs=rs+1;
            M3(rs,:)=M3(n,:);
            M3(rs,7)=i;
            M3(rs,10)=0;
        end
    end
    for n=1:ra
        if M4(n,11)==1
            M4(n,10)=i; %appointing "next node" information to node in rotor
            ra=ra+1;
            M4(ra,:)=M4(n,:);
            M4(ra,7)=i;
            M4(ra,10)=0;
        end
    end
end
%nodes over teeth 2...5
for m=2:5
    %branch below node
    if m~=2 %if not node 2
        i=i+1;
        rs=rs+1;
    end
end

```

```

        M3(rs,:)=M3(rs-1,:);
        M3(rs,10)=M3(rs,7);
        M3(rs,7)=i;
    end
    %tooth branch
    if S(m)==0 %if there is node at tooth end
        for n=1:rs
            if M3(n,11)==m
                rs=rs+1;
                M3(rs,:)=M3(n,:);
                M3(rs,10)=M3(rs,7);
                M3(rs,7)=i;
            end
        end
    else %if there is no node at tooth end
        for n=1:rs
            if M3(n,11)==m
                M3(n,10)=i; %appointing "next node" information to node in rotor
                rs=rs+1;
                M3(rs,:)=M3(n,:);
                M3(rs,7)=i;
                M3(rs,10)=0;
            end
        end
    end
    for n=1:ra
        if M4(n,11)==m
            M4(n,10)=i; %appointing "next node" information to node in rotor
            ra=ra+1;
            M4(ra,:)=M4(n,:);
            M4(ra,7)=i;
            M4(ra,10)=0;
        end
    end
    end
    %branch above node
    j=j+1;
    rs=rs+1;
    if m==5
        M3(rs,2)=8.66;
        M3(rs,12)=1;
    else
        M3(rs,2)=17.32;
    end
    M3(rs,3)=20.16;
    M3(rs,7)=i;
    M3(rs,8)=j;
end
%% solving
G(i,i)=0; %matrix size
J(i)=0; %matrix size
m=0; %aux matrix row counter
aux=zeros(1,15); %aux matrix for finding reluctance of branch with multiple reluctances

for n=1:ra
    M4(n,1)=reluctance_air(M4(n,2),M4(n,3),EL); %airgap reluctance calculation
    M4(n,9)=1/M4(n,1); %permeance calculation

    if M4(n,13)==0 %if not branch with multiple reluctances
        G(M4(n,7),M4(n,7))=G(M4(n,7),M4(n,7))+M4(n,9); %node permeances sum matrix
        if M4(n,10)~=0 %if branch has "next node"
            G(M4(n,7),M4(n,10))=G(M4(n,7),M4(n,10))-M4(n,9); %sum matrix for permeances
        end
    else %if branch with multiple reluctances
        m=m+1;
        aux(m,:)=M4(n,:); %writing to aux matrix
    end

    if S(M4(n,11))==0 %if there is node at tooth end
        if M4(n,13)==0 && M4(n,10)~=0 %if not branch with multiple reluctances and branch
        has "next node" (MMF direction into node)
            J(M4(n,7))=J(M4(n,7))+F(M4(n,11))*M4(n,9); %nodecurrent matrix
        elseif M4(n,13)==0 %if not branch with multiple reluctances and branch has no "next
        node" (MMF direction out of node)
            J(M4(n,7))=J(M4(n,7))-F(M4(n,11))*M4(n,9); %nodecurrent matrix
        end
    end
end
end

```



```

end
G_2=G;
aux_2=aux;
m_2=m;
J_2=J;
e=0; %backup for ending while cycle

%*barrier 1 reluctance R4 is determined for half of the total area
M3(3,3)=M3(3,3)*2;
M3(3,15)=M3(3,15)*2;

while 1
    for n=1:rs
        M3(n,4)=1.5; %initial values for flux densities
    end

    for p=1:150
        G=G_2;
        aux=aux_2;
        m=m_2;
        J=J_2;

        for n=1:rs
            M3(n,6)=mu(M3(n,4)); %finding mu
            if M3(n,15)~=0 && abs(M3(n,3)-M3(n,15))>0.0001 %if is trapezoid reluctance and
bases have difference over 0.0001
                M3(n,1)=reluctance_steel_tr(M3(n,2),M3(n,3),M3(n,15),EL,M3(n,6));
%reluctance calculation for trapezoid
            else
                M3(n,1)=reluctance_steel(M3(n,2),M3(n,3),EL,M3(n,6)); %reluctance
calculation
            end
            M3(n,9)=1/M3(n,1); %permeance calculation

            if M3(n,13)==0 %if not branch with multiple reluctances
                G(M3(n,7),M3(n,7))=G(M3(n,7),M3(n,7))+M3(n,9); %node permeances sum matrix
                if M3(n,10)~=0 %if branch has "next node"
                    G(M3(n,7),M3(n,10))=G(M3(n,7),M3(n,10))-M3(n,9); %sum matrix for
permeances between nodes
                end
            else %if branch with multiple reluctances
                if aux==0 %if aux matrix is empty
                    m=m+1;
                    aux(m,:)=M3(n,:);
                else
                    for k=1:size(aux,1)
                        if M3(n,7)==aux(k,7) && M3(n,10)==aux(k,10) %if node and "next
node" are same
                            aux(k,1)=aux(k,1)+M3(n,1);
                            break
                        end
                        if k==size(aux,1) %if reached last row
                            m=m+1;
                            aux(m,:)=M3(n,:);
                        end
                    end
                end
            end
        end
    end

    for n=1:size(aux,1)
        aux(n,9)=1/aux(n,1); %permeance calculation for aux matrix
        G(aux(n,7),aux(n,7))=G(aux(n,7),aux(n,7))+aux(n,9); %adding aux matrix
permeances to node permeances sum matrix
        if aux(n,10)~=0 %if branch has "next node"
            G(aux(n,7),aux(n,10))=G(aux(n,7),aux(n,10))-aux(n,9); %adding aux matrix
permeances to "sum matrix for permeances between nodes"
        end
        if aux(n,11)~=0 %if branch is in tooth
            if S(aux(n,11))==1 %if there is no node at tooth end (branch contains MMF)
                if aux(n,10)~=0 %if branch has "next node" (MMF direction into node)
                    J(aux(n,7))=J(aux(n,7))+F(aux(n,11))*aux(n,9); %nodecurrent matrix
                else %if branch has no "next node" (MMF direction out of node)
                    J(aux(n,7))=J(aux(n,7))-F(aux(n,11))*aux(n,9); %nodecurrent matrix
                end
            end
        end
    end
end
end

```

```

end

%calculate nodes potentials
G=G+tril(G,-1)'+triu(G,1)';
U=inv(G)*transpose(J);

%calculate fluxes
for n=1:rs
    if M3(n,10)~=0 && M3(n,12)==0 && M3(n,13)==0 %if branch has "next node" and is
not connected with last node and is not branch with multiple reluctances
        M9(M3(n,8))=(U(M3(n,7))-U(M3(n,10)))/M3(n,1);
    end
end
for n=1:size(aux,1)
    if aux(n,10)~=0 && aux(n,12)==0 && aux(n,4)~=0 %if branch has "next node" and
is not connected with last node and has B value (steel reluctance)
        M9(aux(n,8))=(U(aux(n,7))-U(aux(n,10)))/aux(n,1);
    elseif aux(n,10)~=0 && aux(n,12)==0 %if branch has "next node" and has no B
value (branch contains MMF)
        M9(aux(n,8))=(F(aux(n,11))+(U(aux(n,10))-U(aux(n,7))))/aux(n,1);
    end
end
%calculate fluxes for branches connected to last node
for n=1:rs
    if M3(n,12)~=0 && M3(n,13)==0 %if branch has "last" info and is not branch with
multiple reluctances
        M9(M3(n,8))=U(M3(n,7))/M3(n,1);
    end
end
M9(M3(n,8))=-M9(M3(n,8)); %determining flux direction for branches connected to
last node
for n=1:size(aux,1)
    if aux(n,12)~=0 %if branch has "last" info
        M9(aux(n,8))=U(aux(n,7))/aux(n,1);
    end
end

%barrier 1 reluctance R4 is determined for half of the total area
M9(M3(3,8))=M9(M3(3,8))*2;

%calculate "new" flux density B'
for n=1:rs
    if M3(n,15)~=0 %if is trapezoid reluctance
        M3(n,5)=M9(M3(n,8))/(((M3(n,3)+M3(n,15))/2)/1000)*(EL/1000); %average
"new" flux density B'
    else
        M3(n,5)=M9(M3(n,8))/((M3(n,3)/1000)*(EL/1000)); %"new" flux density B'
    end
    error=0.01;
    if M3(n,5)<M3(n,4)
        M3(n,4)=M3(n,4)-error;
    else
        M3(n,4)=M3(n,4)+error;
    end
end

if p==149
    M10(1,:)=M9;
end
if p==150
    M10(2,:)=M9;
end

end

%if flux is negative - change branch "next node" info
neg=0;
for k=1:size(M9,2)
    if M9(k)<0
        neg=1; %at least one negative value
        for n=1:rs
            if M3(n,8)==k && M3(n,10)~=0 && M3(n,13)==0 %if is same branch and branch
has "next node" and is not branch with multiple reluctances
                for t=1:rs
                    if M3(t,8)==k && M3(t,10)==0 && M3(n,13)==0 %if is same branch and
branch has no "next node" and is not branch with multiple reluctances
                        M3(t,10)=M3(n,7);
                        M3(n,10)=0;
                    end
                end
            end
        end
    end
end

```

```

                                break
                            end
                        end
                    end
                end
            end
        end
    end
    if neg==0 || e==10 %if no negative values or cycle number full
        break
    end
    e=e+1;
end
%calculate average flux (necessary because "mu finding" is in steps)
M9=mean(M10,1);

%*barrier 1 reluctance R4 is determined for half of the total area
M9(M3(3,8))=M9(M3(3,8))/2;

%calculate fluxes according to teeth
for m=1:size(F,2)
    for n=1:rs
        if M3(n,11)==m && M3(n,10)~=0 %if branch is in tooth and has "next node"
            v_D(m)=M9(M3(n,8));
            break
        end
    end
end
v_D=diag(v_D);
%% Q-AXIS
%% rearrange matrices for q-axis
j=0;
for i=nb:-1:1
    j=j+1;
    MI_2(j,:)=MBI(i,:);
    MPX_2(j,:)=MPX(i,:);
    MPY_2(j,:)=MPY(i,:);
end
MBI=MI_2;
MPX=MPX_2;
MPY=MPY_2;
%barrier B7 angle recalculation
for i=1:nb
    MBI(i,7)=45-MBI(i,7);
end
%MBA - matrix barrier angles - barrier b end lower edge, middle, upper edge
for b=1:nb
    MBA(b,2)=MBI(b,7);
    MBA(b,1)=MBI(b,7)-MBI(b,6)/2;
    MBA(b,3)=MBI(b,7)+MBI(b,6)/2;
end
%% declaring variables
%info - [reluctance,length,width,B,B',mu,i,j,permeance,next node,MMF,1-connected to last
node,1-multiple reluctances,1-radial bridge]
M3=zeros(1,14); %steel reluctance info matrix
M4=zeros(1,14); %air reluctance info matrix
i=0; %node counter
j=0; %branch (flux) counter
b=0; %barrier counter
rs=0; %steel reluctance counter
ra=0; %air reluctance counter
%% first node
i=i+1;
b=b+1;
for n=1:5 %finding barrier
    if MBA(b,1)<MTA(n,1) %if barrier lower edge is on lower slot
        %tangential bridge above node
        j=j+1;
        rs=rs+1;
        if MBA(b,3)<MTA(n,1) %if barrier upper edge is on lower slot
            M3(rs,2)=(MBA(b,3)-MBA(b,1))*(Dro*pi/360);
            if b==nb %if last barrier
                M3(rs,12)=1;
            end
        elseif MBA(b,3)<MTA(n,2) %if barrier upper edge is on tooth
            M3(rs,2)=(MBA(b,3)-MTA(n,1))/2+MTA(n,1)-MBA(b,1)*(Dro*pi/360);
        else %if barrier upper edge is above tooth

```

```

        M3(rs,2)=(MTA(n,2)-MTA(n,1))/2+MTA(n,1)-MBA(b,1))*(Dro*pi/360);
    end
    break
end
if MBA(b,1)<MTA(n,2) %if barrier lower edge is on tooth
    %airgap reluctance and MMF
    j=j+1;
    ra=ra+1;
    M4(ra,2)=ag;
    M4(ra,3)=(MBA(b,1)-MTA(n,1))*(Dsi*pi/360);
    M4(ra,7)=i;
    M4(ra,8)=j;
    M4(ra,11)=n;

    %tangential bridge above node
    j=j+1;
    rs=rs+1;
    if MBA(b,3)<MTA(n,2) %if barrier upper edge is on tooth
        M3(rs,2)=(MBA(b,2)-MBA(b,1))*(Dro*pi/360);
    else %if barrier upper edge is above tooth
        M3(rs,2)=(MTA(n,2)-MBA(b,1))/2*(Dro*pi/360);
    end
    break
end
%airgap reluctance and MMF
j=j+1;
ra=ra+1;
M4(ra,2)=ag;
M4(ra,3)=(MTA(n,2)-MTA(n,1))*(Dsi*pi/360);
M4(ra,7)=i;
M4(ra,8)=j;
M4(ra,11)=n;
end
M3(rs,3)=MBI(b,4);
M3(rs,7)=i;
M3(rs,8)=j;
M3(rs,10)=i+1;
%barrier above node
j=j+1;
ra=ra+1;
[M4(ra,2),M4(ra,3)]=barrier_eq(MPX,MPY,b);
M4(ra,7)=i;
M4(ra,8)=j;
%radial bridge above node
j=j+1;
rs=rs+1;
M3(rs,2)=MBI(b,5);
M3(rs,3)=MBI(b,2)/2;
M3(rs,7)=i;
M3(rs,8)=j;
M3(rs,14)=1;
%determine next node for barrier and radial bridge
if MBA(b,3)<MTA(n,1) %if whole barrier is on lower slot of tooth
    M4(ra,10)=i+1;
    M3(rs,10)=i+1;
else %if barrier is on tooth
    M4(ra,10)=i+2;
    M3(rs,10)=i+2;
end
if n~=5
    %finding upper edge of barrier
    p=1; %aux variable
    for u=n:4
        if MBA(b,3)>MTA(u+1,1) %if barrier is above tooth u+1 lower edge
            M4(ra,10)=i+2+p;
            M3(rs,10)=i+2+p;
        else
            break
        end
        p=p+1;
    end
end
if b==nb %if last barrier
    M4(ra,12)=1;
    M3(rs,12)=1;
end
end
%% intermediate nodes

```

```

for t=1:5
    while 1 %repeat if there are more than one barriers on tooth or on above slot
        if MBA(b,3)>MTA(t,1) && MBA(b,1)<MTA(t,2) %if barrier is on tooth
            %node on barrier end
            %tangnetial bridge below node
            i=i+1;
            rs=rs+1;
            if t==1
                M3(rs,:)=M3(rs-2,:);
            else
                if MBA(b,1)>MTA(t-1,2) %if barrier lower edge is on tooth or on lower
slot
                    M3(rs,:)=M3(rs-2,:);
                else %if barrier is on lower tooth
                    M3(rs,:)=M3(rs-1,:);
                end
            end
            M3(rs,7)=i;
            M3(rs,10)=0;

            %airgap reluctance and MMF
            j=j+1;
            ra=ra+1;
            M4(ra,2)=ag;
            if MBA(b,1)<MTA(t,1) && MBA(b,3)<MTA(t,2) %if barrier lower edge is under
tooth and barrier upper edge is on tooth
                M4(ra,3)=(MBA(b,3)-MTA(t,1))*(Dsi*pi/360);
            elseif MBA(b,1)<MTA(t,1) && MBA(b,3)>MTA(t,2) %if barrier lower edge is
under tooth and barrier upper edge is above tooth
                M4(ra,3)=(MTA(t,2)-MTA(t,1))*(Dsi*pi/360);
            elseif MBA(b,1)>MTA(t,1) && MBA(b,3)<MTA(t,2) %if barrier lower and upper
edge is on tooth
                M4(ra,3)=(MBA(b,3)-MBA(b,1))*(Dsi*pi/360);
            else %if barrier lower edge is on tooth and barrier upper edge is above
tooth
                M4(ra,3)=(MTA(t,2)-MBA(b,1))*(Dsi*pi/360);
            end
            M4(ra,7)=i;
            M4(ra,8)=j;
            M4(ra,11)=t;

            %tangential bridge above node
            j=j+1;
            rs=rs+1;
            if MBA(b,3)<MTA(t,2) || t==5 %if barrier upper edge is on tooth or if last
tooth
                M3(rs,2)=M4(ra,3)/2;
                if b==nb %if last barrier
                    M3(rs,12)=1;
                end
                elseif MBA(b,3)<MTA(t+1,1) %if barrier upper edge is on upper slot
                    M3(rs,2)=(MBA(b,3)-MTA(t,2))*(Dro*pi/360)+M4(ra,3)/2;
                    if b==nb %if last barrier
                        M3(rs,12)=1;
                    end
                elseif MBA(b,3)<MTA(t+1,2) %if barrier upper edge is on upper tooth
                    M3(rs,2)=((MBA(b,3)-MTA(t+1,1))/2+MTA(t+1,1)-
MTA(t,2))*(Dro*pi/360)+M4(ra,3)/2;
                else %if barrier upper edge is above upper tooth
                    M3(rs,2)=((MTA(t+1,2)-MTA(t+1,1))/2+MTA(t+1,1)-
MTA(t,2))*(Dro*pi/360)+M4(ra,3)/2;
                end
                M3(rs,3)=MBI(b,4);
                M3(rs,7)=i;
                M3(rs,8)=j;
                M3(rs,10)=i+1;
            end

            if t~=5 %if not last tooth
                if MBA(b,3)>MTA(t+1,1) %if barrier upper edge is above upper tooth lower edge
                    break %next tooth
                end
            end

            if b==nb %if last barrier
                break
            end
        end
    end
end

```

```

b=b+1;
%node between barriers
i=i+1;
%tangential bridge below node
rs=rs+1;
if MBA(b-1,3)>MTA(t,1) && MBA(b-1,1)<MTA(t,2) %if previous barrier is on tooth
    M3(rs,:)=M3(rs-1,:);
else
    M3(rs,:)=M3(rs-2,:);
end
M3(rs,7)=i;
M3(rs,10)=0;
%barrier and radial bridge below node
ra=ra+1;
rs=rs+1;
if MBA(b-1,1)>MTA(t,2) %if whole previous barrier is on upper slot
    M4(ra,:)=M4(ra-1,:);
    M3(rs,:)=M3(rs-2,:);
else %if previous barrier is on tooth
    M4(ra,:)=M4(ra-2,:);
    M3(rs,:)=M3(rs-4,:);
end
if t~=1
    %finding lower edge of barrier
    p=1; %aux variable
    for n=t-1:-1:1
        if MBA(b-1,1)>MTA(n,2) %if previous barrier lower edge is above tooth
            break
        else
            M4(ra,:)=M4(ra-2-p,:);
            M3(rs,:)=M3(rs-4-2*p,:);
        end
        p=p+1;
    end
end
M4(ra,7)=i;
M4(ra,10)=0;
M3(rs,7)=i;
M3(rs,10)=0;

upper edge
for n=t:5 %finding barrier
    if MBA(b,1)<MTA(n,1) %if barrier lower edge is on lower slot
        %tangential bridge above node
        j=j+1;
        rs=rs+1;
        if MBA(b,3)<MTA(n,1) %if barrier upper edge is on lower slot
            M3(rs,2)=(MBA(b,3)-MBA(b,1))*(Dro*pi/360);
            if b==nb %if last barrier
                M3(rs,12)=1;
            end
        elseif MBA(b,3)<MTA(n,2) %if barrier upper edge is on tooth
            M3(rs,2)=(MBA(b,3)-MTA(n,1))/2+MTA(n,1)-MBA(b,1))*(Dro*pi/360);
        else %if barrier upper edge is above tooth
            M3(rs,2)=(MTA(n,2)-MTA(n,1))/2+MTA(n,1)-MBA(b,1))*(Dro*pi/360);
        end
        break
    end
    if MBA(b,1)<MTA(n,2) %if barrier lower edge is on tooth
        %airgap reluctance and MMF
        j=j+1;
        ra=ra+1;
        M4(ra,2)=ag;
        if n==t %if barrier is on tooth t
            M4(ra,3)=(MBA(b,1)-MBA(b-1,2))*(Dsi*pi/360);
        else %if barrier lower edge is on tooth t
            M4(ra,3)=(MBA(b,1)-MTA(n,1))*(Dsi*pi/360);
        end
        M4(ra,7)=i;
        M4(ra,8)=j;
        M4(ra,11)=n;
        %tangential bridge above node
        j=j+1;
        rs=rs+1;
        if MBA(b,3)<MTA(n,2) %if barrier upper edge is on tooth
            M3(rs,2)=(MBA(b,2)-MBA(b,1))*(Dro*pi/360);

```

```

else %if barrier upper edge is above tooth
    M3(rs,2)=(MTA(n,2)-MBA(b,1))/2)*(Dro*pi/360);
end
break
end

if n==t
    if MBA(b-1,3)<MTA(t,2) %if previous barrier upper edge is on tooth t
        %airgap reluctance and MMF
        j=j+1;
        ra=ra+1;
        M4(ra,2)=ag;
        M4(ra,3)=(MTA(t,2)-MBA(b-1,3))*(Dsi*pi/360);
        M4(ra,7)=i;
        M4(ra,8)=j;
        M4(ra,11)=n;
    end
else
    %airgap reluctance and MMF
    j=j+1;
    ra=ra+1;
    M4(ra,2)=ag;
    M4(ra,3)=(MTA(n,2)-MTA(n,1))*(Dsi*pi/360);
    M4(ra,7)=i;
    M4(ra,8)=j;
    M4(ra,11)=n;
end
end
M3(rs,3)=MBI(b,4);
M3(rs,7)=i;
M3(rs,8)=j;
M3(rs,10)=i+1;

%barrier above node
j=j+1;
ra=ra+1;
[M4(ra,2),M4(ra,3)]=barrier_eq(MPX,MPY,b);
M4(ra,7)=i;
M4(ra,8)=j;
%radial bridge above node
j=j+1;
rs=rs+1;
M3(rs,2)=MBI(b,5);
M3(rs,3)=MBI(b,2)/2;
M3(rs,7)=i;
M3(rs,8)=j;
M3(rs,14)=1;
%determine next node for barrier and radial bridge
if MBA(b,3)<MTA(n,1) %if whole barrier is on lower slot of tooth
    M4(ra,10)=i+1;
    M3(rs,10)=i+1;
else %if barrier is on tooth
    M4(ra,10)=i+2;
    M3(rs,10)=i+2;
end
if n~=5
    %finding upper edge of barrier
    p=1; %aux variable
    for u=n:4
        if MBA(b,3)>MTA(u+1,1) %if barrier is above tooth u+1 lower edge
            M4(ra,10)=i+2+p;
            M3(rs,10)=i+2+p;
        else
            break
        end
        p=p+1;
    end
end
if b==nb %if last barrier
    M4(ra,12)=1;
    M3(rs,12)=1;
end
end

end
%% remaining air gaps
for n=5:-1:1
    if MBA(b,3)>MTA(n,1) %if barrier upper edge is above tooth lower edge

```

```

    if MBA(b,3)<MTA(n,2) %if barrier upper edge is on tooth
        %airgap reluctance and MMF
        j=j+1;
        ra=ra+1;
        M4(ra,2)=ag;
        M4(ra,3)=(MTA(n,2)-MBA(b,3))*(Dsi*pi/360);
        M4(ra,8)=j;
        M4(ra,11)=n;
        M4(ra,12)=1;
    end
    break
else
    %airgap reluctance and MMF
    j=j+1;
    ra=ra+1;
    M4(ra,2)=ag;
    M4(ra,3)=(MTA(n,2)-MTA(n,1))*(Dsi*pi/360);
    M4(ra,8)=j;
    M4(ra,11)=n;
    M4(ra,12)=1;
end
end
%% solving
G=0;
G(i,i)=0; %matrix size
J=0;
J(i)=0; %matrix size

for n=1:ra
    M4(n,1)=reluctance_air(M4(n,2),M4(n,3),EL); %airgap reluctance calculation
    M4(n,9)=1/M4(n,1); %permeance calculation
    if M4(n,11)==0 || M4(n,12)==0 %if branch contains no MMF or branch has no "last" info
        G(M4(n,7),M4(n,7))=G(M4(n,7),M4(n,7))+M4(n,9); %node permeances sum matrix
    end
    if M4(n,10)~=0 && M4(n,12)==0 %if branch has "next node" and branch has no "last" info
        G(M4(n,7),M4(n,10))=G(M4(n,7),M4(n,10))-M4(n,9); %sum matrix for permeances between
nodes
    end
    if M4(n,11)~=0 && M4(n,12)==0 %if branch contains MMF or branch has no "last" info
        J(M4(n,7))=J(M4(n,7))+F(M4(n,11))*M4(n,9); %nodecurrent matrix
    end
end
end
G_2=G;
e=0; %while tsükli lõpetamise backup
%barrier radial bridge reluctance is determined for half of the total area
for n=1:rs
    if M3(n,14)~=0
        M3(n,3)=M3(n,3)*2;
    end
end
end

while 1
    for n=1:rs
        M3(n,4)=1.5; %initial values for flux densities
    end

    for p=1:150
        G=G_2;
        for n=1:rs
            M3(n,6)=mu(M3(n,4)); %finding mu
            M3(n,1)=reluctance_steel(M3(n,2),M3(n,3),EL,M3(n,6)); %reluctance calculation
            M3(n,9)=1/M3(n,1); %permeance calculation

            G(M3(n,7),M3(n,7))=G(M3(n,7),M3(n,7))+M3(n,9); %node permeances sum matrix
            if M3(n,10)~=0 && M3(n,12)==0 %if branch has "next node" and branch has no
"last" info
                G(M3(n,7),M3(n,10))=G(M3(n,7),M3(n,10))-M3(n,9); %sum matrix for permeances
between nodes
            end
        end
        end

        %calculate nodes potentials
        G=G+tril(G',-1);
        U=inv(G)*transpose(J);
        %calculate fluxes
        for n=1:rs

```



```

        if M3(n,10)~=0 && M3(n,12)==0 %if branch has "next node" and branch has no
"last" info
            M12(M3(n,8))=(U(M3(n,7))-U(M3(n,10)))/M3(n,1);
            if M3(n,14)~=0 %if branch is for radial bridge
                M12(M3(n,8))=M12(M3(n,8))*2; %*barrier radial bridge reluctance is
determined for half of the total area
            end
        end
    end
    %flux calculation for branches connected to last node (base node)
    for n=1:rs %voogude arvutamine
        if M3(n,12)~=0
            M12(M3(n,8))=U(M3(n,7))/M3(n,1);
            if M3(n,14)~=0 %if branch is for radial bridge
                M12(M3(n,8))=M12(M3(n,8))*2; %*barrier radial bridge reluctance is
determined for half of the total area
            end
        end
    end

    %calculate "new" flux density B'
    for n=1:rs
        M3(n,5)=M12(M3(n,8))/(M3(n,3)/1000)*(EL/1000); %"new" flux density B'
        error=0.01;
        if M3(n,5)<M3(n,4)
            M3(n,4)=M3(n,4)-error;
        else
            M3(n,4)=M3(n,4)+error;
        end
    end

    if p==149
        M11(1,:)=M12;
    end
    if p==150
        M11(2,:)=M12;
    end

    %if flux is negative - change branch "next node" info
    neg=0;
    for k=1:size(M12,2)
        if M12(k)<0
            neg=1; %at least one negative value
            for n=1:rs
                if M3(n,8)==k && M3(n,10)~=0 %if is same branch and branch has "next node"
                    for t=1:rs
                        if M3(t,8)==k && M3(t,10)==0 %if is same branch and branch has no
"next node"
                            M3(t,10)=M3(n,7);
                            M3(n,10)=0;
                            break
                        end
                    end
                end
            end
        end
    end
    if neg==0 || e==10 %if no negative values or cycle number full
        break
    end
    e=e+1;
end
%calculate average flux (necessary because "mu finding" is in steps)
M12=mean(M11,1);
%*barrier radial bridge reluctance is determined for half of the total area
for n=1:rs
    if M3(n,14)~=0 && M3(n,10)~=0
        M12(M3(n,8))=M12(M3(n,8))/2;
    end
end
%air gap branch flux calculation
for n=1:ra
    if M4(n,11)~=0 && M4(n,12)==0
        M12(M4(n,8))=(F(M4(n,11))-U(M4(n,7)))/M4(n,1);
    end
end

```

```

end
%remaining air gap branch flux calculation
for n=1:ra
    if M4(n,11)~=0 && M4(n,12)~=0
        M12(M4(n,8))=F(M4(n,11))/M4(n,1);
    end
end

%additional last tooth leakage flux (in slot)
leakage=F(5)/reluctance_air(1,1,EL);

%calculate fluxes according to teeth
v_Q(size(F,2))=0;
for n=1:ra
    if M4(n,11)~=0
        v_Q(M4(n,11))=v_Q(M4(n,11))+M12(M4(n,8));
    end
end
v_Q(5)=v_Q(5)+leakage; %additional leakage flux
v_Q=diag(v_Q);
%% inductance calculation
%phase currents average value
Iavg=(abs(IA)+abs(IB)+abs(IC))/PH;
%WPI - winding placement matrix for inductance calculation - [tooth1 tooth2 ...]
if IA>=0
    WPI(1,:)=WP(1,:);
else
    WPI(1,:)=-1*WP(1,:);
end
if IB>=0
    WPI(2,:)=WP(2,:);
else
    WPI(2,:)=-1*WP(2,:);
end
if IC>=0
    WPI(3,:)=WP(3,:);
else
    WPI(3,:)=-1*WP(3,:);
end
%sum matrix for number of turns (per tooth)
M21=diag(sum(WPI*NT));
%inductance calculation
M22=(M21*v_D)/Iavg;
L_D=sum(M22,'all')*8*kW
M22=(M21*v_Q)/Iavg;
L_Q=sum(M22,'all')*8*kW

```

L2.2 Programmikoodi funktsioonid

```

function [length,width] = barrier_eq(MPX,MPY,b)
    %barrier parallel reluctances (R1,R2) equivalent length, width
    R1_length=MPY(b,2)-MPY(b,1);
    R1_width=MPX(b,3)-MPX(b,8);
    R2_length=sqrt((MPX(b,12)-MPX(b,11))^2+(MPY(b,12)-MPY(b,11))^2);
    R2_width=sqrt((MPX(b,9)-MPX(b,3))^2+(MPY(b,9)-MPY(b,3))^2);
    length=R1_length*R2_length;
    width=R1_width*R2_length+R2_width*R1_length;
end


---


function [R5_length,R5_width] = barrier_last(MPX,MPY,b,Dro)
    %last barrier upper reluctance (R5) length, width
    R5_length=MPX(b,5);
    R5_width=Dro/2-MPY(b,5);
end


---


function [R1_length,R1_width,R2_length,R2_width,R2_width2] = barrier_lower(MPX,MPY,k)
    %barrier lower reluctances (R1,R2) length, width
    R1_width=MPY(k,1)-MPY(k-1,2);
    XB=(MPX(k-1,5)-MPX(k,4))/2+MPX(k,4);
    YB=(MPY(k-1,5)-MPY(k,4))/2+MPY(k,4);
    R1_length=XB;
    XC=(MPX(k-1,14)-MPX(k,13))/2+MPX(k,13);
    YC=(MPY(k,13)-MPY(k-1,14))/2+MPY(k-1,14);
    R2_length=sqrt((XC-XB)^2+(YC-YB)^2);
    t=- (MPX(k-1,7)-MPX(k-1,5))/(MPY(k-1,7)-MPY(k-1,5)); %slope between (lower barrier)
    point 5 and D
    XD=(-MPY(k-1,5)*MPX(k,4)+t*MPX(k-1,5)*MPX(k,4)+MPY(k,6)*MPX(k,4)+MPY(k-1,5)*MPX(k,6)-
    t*MPX(k-1,5)*MPX(k,6)-MPX(k,6)*MPY(k,4))/(t*MPX(k,4)-t*MPX(k,6)-MPY(k,4)+MPY(k,6));
    YD=MPY(k-1,5)+t*XD-t*MPX(k-1,5);
    XE=(-MPY(k-1,7)*MPX(k,4)+t*MPX(k-1,7)*MPX(k,4)+MPY(k,6)*MPX(k,4)+MPY(k-1,7)*MPX(k,6)-
    t*MPX(k-1,7)*MPX(k,6)-MPX(k,6)*MPY(k,4))/(t*MPX(k,4)-t*MPX(k,6)-MPY(k,4)+MPY(k,6));
    YE=MPY(k-1,7)+t*XE-t*MPX(k-1,7);
    R2_width=sqrt((MPX(k-1,5)-XD)^2+(MPY(k-1,5)-YD)^2);
    R2_width2=sqrt((MPX(k-1,7)-XE)^2+(MPY(k-1,7)-YE)^2);
end


---


function [R3_length,R3_width,R4_length,R4_width,R4_width2] = barrier_lower_first(MPX,MPY)
    %first barrier lower reluctances (R3,R4) length, width
    b=1;
    R3_width=MPY(b,1)-16;
    R3_length=MPX(b,4);
    r4=sqrt(MPX(b,4)^2+MPY(b,4)^2);
    XG=r4*cos((45*pi)/180);
    YG=XG;
    r20=sqrt(MPX(b,13)^2+MPY(b,13)^2);
    XH=r20*cos((45*pi)/180);
    YH=XH;
    R4_length=sqrt((XH-XG)^2+(YH-YG)^2);
    R4_width=sqrt((XG-MPX(b,4))^2+(YG-MPY(b,4))^2);
    t=-1; %slope between H and I
    XI=(-YH*MPX(b,4)+t*XH*MPX(b,4)+MPY(b,6)*MPX(b,4)+YH*MPX(b,6)-t*XH*MPX(b,6)-
    MPX(b,6)*MPY(b,4))/(t*MPX(b,4)-t*MPX(b,6)-MPY(b,4)+MPY(b,6));
    YI=YH+t*XI-t*XH;
    R4_width2=sqrt((XH-XI)^2+(YH-YI)^2);
end


---


function [out] = mu(B)
    A=[ %B values
    0.0001
    0.01
    0.025
    0.0375
    0.05
    0.0625
    0.075
    0.0875
    0.1
    0.1125
    0.125
    0.1375
    0.15
    0.1625
    0.175
    0.1875
    0.2
    0.2125
    0.225

```

0.2375
0.25
0.2625
0.275
0.2875
0.3
0.3125
0.325
0.3375
0.35
0.3625
0.375
0.3875
0.4
0.4125
0.425
0.4375
0.45
0.4625
0.475
0.4875
0.5
0.5125
0.525
0.5375
0.55
0.5625
0.575
0.5875
0.6
0.6125
0.625
0.6375
0.65
0.6625
0.675
0.6875
0.7
0.7125
0.725
0.7375
0.75
0.7625
0.775
0.7875
0.8
0.8125
0.825
0.8375
0.85
0.8625
0.875
0.8875
0.9
0.9125
0.925
0.9375
0.95
0.9625
0.975
0.9875
1
1.0125
1.025
1.0375
1.05
1.0625
1.075
1.0875
1.1
1.1125
1.125
1.1375
1.15
1.1625
1.175
1.1875

1.2
1.2125
1.225
1.2375
1.25
1.2625
1.275
1.2875
1.3
1.3125
1.325
1.3375
1.35
1.3625
1.375
1.3875
1.4
1.4125
1.425
1.4375
1.45
1.4625
1.475
1.4875
1.5
1.5125
1.525
1.5375
1.55
1.5625
1.575
1.5875
1.6
1.6125
1.625
1.6375
1.65
1.6625
1.675
1.6875
1.7
1.7125
1.725
1.7375
1.75
1.7625
1.775
1.7875
1.8
1.8125
1.825
1.8375
1.85
1.8625
1.875
1.8875
1.9
1.9125
1.925
1.9375
1.95
1.9625
1.975
1.9875
2
2.0125
2.025
2.0375
2.05
2.0625
2.075
2.0875
2.1
2.1125
2.125
2.1375
2.15

```

2.1625
2.175
2.1875
2.2
2.2125
2.225
2.2375
2.25
2.2625
2.275
2.2875
2.3
2.3125
2.325
2.3375
2.35
2.3625
2.375
2.3875
2.4
2.4125
2.425
2.4375
2.45
2.4625
2.475
2.4875
2.5
2.55
2.6
2.65
2.7
2.75
2.8
2.85
2.9
2.95
3
3.05
3.1
3.15
3.2
3.25
3.3
3.35
3.4
3.45
3.5
3.55
3.6
3.65
3.7
3.75
3.8
3.85
3.9
3.95
4
];

AA=[ %mu values
1
994.7160682
1530.332413
1761.609609
1992.886806
2159.887126
2326.887446
2493.887767
2660.888087
2815.514682
2970.141276
3124.767871
3279.394466
3421.282315
3563.170165
3705.058014

```

3846.945863
3976.139812
4105.33376
4234.527709
4363.721657
4480.340749
4596.959842
4713.578934
4830.198026
4934.388045
5038.578064
5142.768083
5246.958102
5338.879647
5430.801193
5522.722738
5614.644283
5694.468578
5774.292872
5854.117166
5933.94146
6001.848371
6069.755283
6137.662194
6205.569106
6261.745497
6317.921888
6374.098279
6430.27467
6474.912564
6519.550458
6564.188351
6608.826245
6642.120147
6675.414048
6708.70795
6742.001852
6764.144943
6786.288035
6808.431127
6830.574218
6841.752439
6852.930661
6864.108882
6875.287103
6875.67057
6876.054037
6876.437505
6876.820972
6866.550888
6856.280805
6846.010721
6835.740638
6814.910064
6794.079491
6773.248918
6752.418345
6721.043561
6689.668778
6658.293994
6626.91921
6584.897641
6542.876073
6500.854504
6458.832935
6405.882367
6352.931799
6299.981231
6247.030663
6182.604045
6118.177427
6053.750809
5989.324192
5912.49847
5835.672748
5758.847026
5682.021304

5591.375229
5500.729153
5410.083078
5319.437003
5212.982736
5106.52847
5000.074204
4893.619937
4768.976337
4644.332737
4519.689137
4395.045537
4250.27896
4105.512383
3960.745805
3815.979228
3651.780971
3487.582714
3323.384457
3159.186199
2982.594596
2806.002992
2629.411388
2452.819785
2280.111025
2107.402266
1934.693506
1761.984747
1614.868218
1467.75169
1320.635161
1173.518633
1067.019608
960.5205834
854.0215587
747.5225339
681.0358636
614.5491932
548.0625228
481.5758524
442.9134049
404.2509575
365.5885101
326.9260627
304.8848134
282.8435641
260.8023148
238.7610656
224.1738153
209.5865651
194.9993149
180.4120646
170.1443246
159.8765845
149.6088444
139.3411043
131.809677
124.2782497
116.7468224
109.215395
103.4992759
97.78315678
92.06703766
86.35091853
81.74962268
77.14832683
72.54703099
67.94573514
63.61145324
59.27717134
54.94288944
50.60860754
46.08974252
41.5708775
37.05201248
32.53314746
29.45707052


```

26.38099357
23.30491663
20.22883969
18.78392808
17.33901647
15.89410486
14.44919325
13.63230082
12.8154084
11.99851598
11.18162355
10.64615125
10.11067895
9.575206653
9.039734354
8.714352015
8.388969677
8.063587338
7.738205
7.488819795
7.23943459
6.990049385
6.74066418
6.5552239
6.369783619
6.184343338
5.998903058
5.855608295
5.712313533
5.569018771
5.425724009
4.969520276
4.597798715
4.28908081
4.028600076
3.805870174
3.613238907
3.444991345
3.296773255
3.165208882
3.04764072
2.941948129
2.846418288
2.759652652
2.680498037
2.60799507
2.541339116
2.479850291
2.422950184
2.37014361
2.321004159
2.275162658
2.232297878
2.192128996
2.154409436
2.118921802
2.085473688
2.053894183
2.024030956
1.995747794
1.968922527
];

smallesterror=1;
smallestplace=231;
for i=1:231
    error=abs(B-A(i));
    if error<smallesterror
        smallesterror=error;
        smallestplace=i;
    end
end
out=AA(smallestplace);
end
function [out]=reluctance_air(length,width,EL)
    out=length/(0.0000125664*width*(EL/1000)*1*1.12);

```

```

end
function [out]=reluctance_steel(length,width,EL,mu)
    out=length/(0.00000125664*width*(EL/1000)*mu);
end
function [out]=reluctance_steel_tr(length,width,width2,EL,mu)
    out=(length/(0.00000125664*(EL/1000)*mu*(width-width2)))*log(width/width2);
end
function [M3,rs] = tooth(M3,rs,i,j,t)
    %stator tooth reluctances (R1,R2,R3) length, width
    rs=rs+1;
    M3(rs,2)=3.64;
    M3(rs,3)=8.96;
    M3(rs,7)=i;
    M3(rs,8)=j;
    M3(rs,11)=t;
    M3(rs,13)=1;
    rs=rs+1;
    M3(rs,2)=15;
    M3(rs,3)=6.53;
    M3(rs,7)=i;
    M3(rs,8)=j;
    M3(rs,11)=t;
    M3(rs,13)=1;
    rs=rs+1;
    M3(rs,2)=1.99;
    M3(rs,3)=8.13;
    M3(rs,7)=i;
    M3(rs,8)=j;
    M3(rs,11)=t;
    M3(rs,13)=1;
end
function [M3,rs] = tooth_first(M3,rs,i,j,t)
    %stator yoke reluctance
    rs=rs+1;
    M3(rs,2)=18;
    M3(rs,3)=20.3;
    M3(rs,7)=i;
    M3(rs,8)=j;
    M3(rs,11)=t;
    M3(rs,13)=1;
    %stator first tooth reluctances (R1,R2,R3) length, width
    rs=rs+1;
    M3(rs,2)=3.64;
    M3(rs,3)=4.48;
    M3(rs,7)=i;
    M3(rs,8)=j;
    M3(rs,11)=t;
    M3(rs,13)=1;
    rs=rs+1;
    M3(rs,2)=15;
    M3(rs,3)=3.26;
    M3(rs,7)=i;
    M3(rs,8)=j;
    M3(rs,11)=t;
    M3(rs,13)=1;
    rs=rs+1;
    M3(rs,2)=1.99;
    M3(rs,3)=4.07;
    M3(rs,7)=i;
    M3(rs,8)=j;
    M3(rs,11)=t;
    M3(rs,13)=1;
end

```

L2.3 Optimeerimise piirangufunktsioon

```

function [c,ceq] = constraint(dim)
    %% machine dimensions
    Dri=45; %inner diameter of rotor
    Dro=135.2; %outer diameter of rotor
    %% barrier dimensions
    t1=0.2; %parallel coefficient for reluctance R2 between barriers
    t2=0.4; %parallel coefficient for reluctance R4 below first barrier
    nb=4; %number of barriers (per pole)
    MI=[ %matrix barrier information - [B1 B2 B3 B4 B5 B6 B7 B8]
    dim(1) dim(5) dim(9) dim(13) dim(17) dim(21) dim(25) dim(29)
    dim(2) dim(6) dim(10) dim(14) dim(18) dim(22) dim(26) dim(30)
    dim(3) dim(7) dim(11) dim(15) dim(19) dim(23) dim(27) dim(31)
    dim(4) dim(8) dim(12) dim(16) dim(20) dim(24) dim(28) dim(32)];
    %% barrier locations and constraints
    ceq=0; %if input dimensions are suitable then ceq=0
    %MPX - matrix placement x-axis - X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14
    %MPY - matrix placement y-axis - Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10 Y11 Y12 Y13 Y14
    for b=1:nb
        MPX(b,1)=MI(b,2)/2;
        MPY(b,1)=Dri/2+MI(b,1)-MI(b,5)/2;
        MPX(b,2)=MPX(b,1);
        MPY(b,2)=Dri/2+MI(b,1)+MI(b,5)/2;
        MPX(b,3)=MPX(b,1)+MI(b,3);
        MPY(b,3)=MPY(b,1)+MI(b,5)/2;
        MPY(b,4)=MPY(b,1);
        MPX(b,4)=MPX(b,3)+(MPY(b,4)-MPY(b,3))/-tan((MI(b,8)*pi)/180);
        MPY(b,5)=MPY(b,2);
        MPX(b,5)=MPX(b,3)+(MPY(b,5)-MPY(b,3))/-tan((MI(b,8)*pi)/180);
        MPX(b,6)=cos((45+MI(b,7)-MI(b,6)/2)*pi/180)*(Dro/2-MI(b,4));
        MPY(b,6)=sin((45+MI(b,7)-MI(b,6)/2)*pi/180)*(Dro/2-MI(b,4));
        MPX(b,7)=cos((45+MI(b,7)+MI(b,6)/2)*pi/180)*(Dro/2-MI(b,4));
        MPY(b,7)=sin((45+MI(b,7)+MI(b,6)/2)*pi/180)*(Dro/2-MI(b,4));
        MPX(b,8)=MI(b,2)/2;
        MPY(b,8)=MPY(b,3);
        MPX(b,9)=cos((45+MI(b,7))*pi/180)*(Dro/2-MI(b,4));
        MPY(b,9)=sin((45+MI(b,7))*pi/180)*(Dro/2-MI(b,4));
        MPX(b,10)=(MPX(b,9)-MPX(b,3))/2+MPX(b,3);
        MPY(b,10)=(MPY(b,9)-MPY(b,3))/2+MPY(b,3);
        k=- (MPX(b,9)-MPX(b,3))/(MPY(b,9)-MPY(b,3)); %slope between 11, 12
        MPX(b,11)=(-
        MPY(b,10)*MPX(b,4)+k*MPX(b,10)*MPX(b,4)+MPY(b,6)*MPX(b,4)+MPY(b,10)*MPX(b,6)-
        k*MPX(b,10)*MPX(b,6)-MPX(b,6)*MPY(b,4))/(k*MPX(b,4)-k*MPX(b,6)-MPY(b,4)+MPY(b,6));
        MPY(b,11)=MPY(b,10)+k*MPX(b,11)-k*MPX(b,10);
        MPX(b,12)=(-
        MPY(b,10)*MPX(b,5)+k*MPX(b,10)*MPX(b,5)+MPY(b,7)*MPX(b,5)+MPY(b,10)*MPX(b,7)-
        k*MPX(b,10)*MPX(b,7)-MPX(b,7)*MPY(b,5))/(k*MPX(b,5)-k*MPX(b,7)-MPY(b,5)+MPY(b,7));
        MPY(b,12)=MPY(b,10)+k*MPX(b,12)-k*MPX(b,10);
        MPX(b,13)=cos((45+MI(b,7)-MI(b,6)/2)*pi/180)*(Dro/2);
        MPY(b,13)=sin((45+MI(b,7)-MI(b,6)/2)*pi/180)*(Dro/2);
        MPX(b,14)=cos((45+MI(b,7)+MI(b,6)/2)*pi/180)*(Dro/2);
        MPY(b,14)=sin((45+MI(b,7)+MI(b,6)/2)*pi/180)*(Dro/2);
        %constraints
        if b~=1
            n1=MPY(b,1)-MPY(b-1,2);
            n2=MPX(b-1,5)-MPX(b,4);
            n3=atan(MPY(b,6)/MPX(b,6))-atan(MPY(b-1,7)/MPX(b-1,7));
            if n1<=0||n2<=0||n3<=0
                ceq=100; %input is not suitable
                break
            end
            ceq=ceq+t1*abs((MPX(b-1,7)-MPX(b-1,5))/(MPY(b-1,7)-MPY(b-1,5))-(MPX(b,6)-
            MPX(b,4))/(MPY(b,6)-MPY(b,4))); %parallelity of reluctance R2 between barriers
        else %first barrier
            n1=atan(MPY(b,6)/MPX(b,6))-(45*pi)/180;
            n2=atan(MPY(b,4)/MPX(b,4))-(45*pi)/180;
            if n1<=0||n2<=0
                ceq=100; %input is not suitable
                break
            end
            ceq=ceq+t2*abs(1-(MPX(b,6)-MPX(b,4))/(MPY(b,6)-MPY(b,4))); %parallelity of
            reluctance R4 below first barrier
        end
        if b==nb
            n1=(90*pi)/180-atan(MPY(b,7)/MPX(b,7));
            if n1<=0
    
```

```
                ceq=100; %input is not suitable
            break
        end
    end
end
c=[];
end
```