



KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS

Vilius Čeponis

**MATRICINĖS PAVAROS, VEIKIANČIOS ĮTAMPOS TRIKDŽIŲ
KOMPENSAVIMO REŽIME, TYRIMAS**

Baigiamasis magistro projektas

Vadovas

Lekt. Gytis Petrauskas

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS
AUTOMATIKOS KATEDRA

**MATRICINĖS PAVAROS, VEIKIANČIOS ĮTAMPOS TRIKDŽIŲ
KOMPENSAVIMO REŽIME, TYRIMAS**

Baigiamasis magistro projektas
Valdymo technologijos (kodas 621H66001)

Vadovas

(parašas) Lekt. Gytis Petrauskas
(data)

Recenzentas

(parašas)
(data)

Projektą atliko

(parašas) Vilius Čeponis
(data)

KAUNAS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS

(Fakultetas)

Vilius Čeponis

(Studento vardas, pavardė)

Valdymo technologijos (kodas 621H66001)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „MATRICINĖS PAVAROS, VEIKIANČIOS ĮTAMPOS TRIKDŽIŲ
KOMPENSAVIMO REŽIME, TYRIMAS“
AKADEMINIO SAŽININGUMO DEKLARACIJA

20 17 m. Birželio 2 d.
Kaunas

Patvirtinu, kad mano, **Viliaus Čeponio**, baigiamasis projektas tema „MATRICINĖS PAVAROS, VEIKIANČIOS ĮTAMPOS TRIKDŽIŲ KOMPENSAVIMO REŽIME, TYRIMAS“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Čeponis, Vilius. MATRICINĖS PAVAROS, VEIKIANČIOS ĮTAMPOS TRIKDŽIŲ KOMPENSAVIMO REŽIME, TYRIMAS. *Valdymo sistemų magistro* baigiamasis projektas / vadovas Lekt. Gytis Petrauskas; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Automatikos katedra.

Mokslo kryptis ir sritis: Elektros ir elektronikos inžinerija, Technologiniai mokslai

Reikšminiai žodžiai: *matricinis, dažnio, keitiklis, įtampos, trikdžiai, kompensavimas.*

Kaunas, 2017. 49 p.

SANTRAUKA

Matricinis dažnio keitiklis, tai naujo tipo dažnio keitiklis, skirtingai nei tradiciniai, neturintys nuolatinės įtampos grandies su energiją kaupiančiais reaktyviaisiais elementais. Toks keitiklis sudarytas tik iš 9 dvikrypčių elektroninių jungiklių, kurie leidžia, bet kurią įėjimo fazę sujungti su, bet kuria išėjimo faze. Tai yra vienas pagrindinių tokių dažnio keitiklių privalumų, tačiau reaktyviųjų elementų nebuvimas lemia maitinimo įtampos trikdžių tiesioginę įtaką išėjimo įtampai.

Pagal literatūroje pateikiamą statistiką atrinkti keturi dažniausiai pasitaikantys maitinimo įtampos trikdžių tipai. Darbe pasiūlytas modifikuotas dažnio keitiklio algoritmas įtampos trikdžių kompensavimui. Modifikavimo esmė – įtampos vektoriaus formavimo sektorių kampo perslinkimas ir sektoriaus kampo išplėtimas, išėjimo įtampai formuoti panaudojant mažiausiai sutrikdytas įtampas.

Algoritmo modifikacijos patikrinimui sudarytas matricinės pavaros MATLAB Simulink modelis sudarytas iš jėgos, valdymo ir apkrovos modelių. Atlikus modeliavimą gauta, kad modifikuotas algoritmas turi teigiamą įtaką efektinei išėjimo įtampos vertei bei šalutinių harmonikų kiekiui trikdžių metu, kai įėjimo įtampos vertė žymiai sumažėja vienoje fazėje. Trikdžių, kurių metu įtampa sumažėja dviejose ar trijose fazėse, modifikuotas algoritmas turi neigiamą įtaką.

Modelio adekvatumas patikrintas fiziniu matricinio dažnio keitiklio modeliu, įvertinus šalutinių harmonikų kiekį bei efektinę išėjimo įtampą Simulink ir fiziniuose modeliuose, gauta, kad Simulink modelis adekvatus.

Čeponis, Vilius. RESEARCH OF MATRIX DRIVE OPERATING IN VOLTAGE SAGS COMPENSATION MODE: *Master's thesis in Control Systems / supervisor Lect. Gytis Petrauskas. Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of Automation.*

Research area and field: Electrical and Electronics Engineering, Technological Sciences

Key words: *frequency, matrix, converter, voltage, sags, compensation.*

Kaunas, 2017. 49 p.

SUMMARY

Matrix frequency converter is a new type of frequency converters that does not have reactive elements which traditional converters have in a DC stage. Matrix converter is made with 9 bipolar electronical switches that lets connect any input phase with any output phase. This is one of the main advantage of these converters but lack of reactive elements means that any disturbance in input voltage is directly seen in output voltage.

According to statistics from literature, four main types of voltage sags of input voltage are set. In this work modification of matrix converter algorithm is proposed. Algorithm is modified by shifting sector of voltage vector and spreading sector that lets using of least disturbed voltage phases.

Modified algorithm is tested by MATLAB Simulnk model that consists of power, control and motor models. Modeling shows that algorithm modification has positive influence for effective voltage value and total harmonics distortion when sag is in one phase. Algorithm modification has negative influence for effective voltage value and total harmonics distortion when sag is in two or three phases.

Adequacy of model is tested by physical model of matrix converter. By measuring total harmonics distortion and effective voltage value I got that model is adequate.

Turinys

Turinys.....	6
Santrumpos	8
Įvadas.....	9
1. Literatūros apžvalga	11
1.1. Tiesioginis matricinis dažnio keitiklis	11
1.2. Tiesioginio matricinio dažnio keitiklio valdymas veikiant maitinimo įtampos trikdžiams. 19	
1.2.1 Įtampos trikdžių kompensavimas panaudojant papildomą energijos kaupiklį.....	19
1.2.2 Įtampos trikdžių kompensavimas didinant moduliacijos koeficientą.....	20
1.2.3 Įtampos kompensavimas didinant įtampos nuostatos signalą	22
1.3. Įtampos trikdžiai Lietuvos elektros energijos tiekimo linijose.....	22
2. Erdvinio vektoriaus moduliacija trikdžių atveju	23
3. Matlab Simulink modelis	26
3.1. Trikdžius energetinėje sistemoje imituojantis modelis.....	26
3.2. Matricinio keitiklio jėgos grandinių modelis.....	28
3.3. Erdvinio vektoriaus moduliacijos modelis	29
3.4. Matricinio keitiklio apkrovos modelis.....	33
4. Fizinis matricinio dažnio keitiklio modelis	34
4.1. Erdvės vektorių moduliacijos algoritmas	34
4.2. Erdvės vektorių moduliacijos modeliui naudojama įranga.....	34
4.3. Erdvės vektorių moduliacijos programa	36
4.4. Fizinio modelio rezultatai	41
5. Modeliavimo rezultatai.....	41
Išvados.....	44
Literatūros sąrašas	45

Priedai..... 47

Santrumpos

IGBT – dvipolis tranzistorius su izoliuota užtūra (insulated-gate bipolar transistor);

TMDK – tiesioginis matricinis dažnio keitiklis;

EVM – erdvės vektorių moduliacija.

Ivadas

Išradus elektros variklis, atsirado poreikis valdyti jų greitį. Asinchroninių variklių greitis gali būti keičiamas keliais būdais. Greitis gali būti keičiamas - keičiant magnetinio lauko polių porų skaičių, tačiau tokiu būdu greitis yra keičiamas šuoliais. Variklio su faziniu rotoriumi greitį galima keisti keičiant rotoriaus grandinės varžą, t.y. didinant rotoriaus varžą, prie to paties sukimo momento, rotoriaus greitis bus mažesnis, taip greitis keičiamas sklandžiai, tačiau dideli energetiniai nuostoliai. Šiuo metu geriausias būdas keisti asinchroninio variklio greitį yra variklio statoriaus jungimas prie reguliuojamo dažnio įtampos šaltinio – dažnio keitiklio [1].

Patys populiariausi šiuo metu naudojami dažnio keitikliai yra sudaryti iš lygintuvo, sudaryto iš diodų tiltelio, tarpinės nuolatinės įtampos grandies, sudarytos iš kondensatorių, bei inverterio, kuris impulso pločio moduliacijos būdu generuoja reikiamo dažnio įtampą. Tokių dažnio keitiklių pagrindinis trūkumas yra tarpinė nuolatinės įtampos grandis, joje naudojami elektrolitiniai kondensatoriai užima apie 30-50% dažnio keitiklio vietos, taip pat minėti elementai turi trumpiausią tarnavimo laiką iš visų dažnio keitikliuose naudojamų elementų. Dar vienas tokių dažnio keitiklių trūkumas yra veikimo aplinkos temperatūra, kurią riboja kondensatorių naudojimas [2].

Dažnio keitiklį neturintį prieš tai minėtų trūkumų pasiūlė mokslininkai Venturini ir Alesina 1980 metais. Tokie dažnio keitikliai vadinami matriciniais, jie sudaryti iš $m \times n$ elektroninių jėgos jungiklių, kurie sujungia m -fazių įtampos šaltinį su n -fazių apkrova. Trifaziame tinkle reikalingi 9 dvikrypčiai elektroniniai jungikliai. [2]. Dėl nepakankamo elektronikos elementų išvystymo tuo metu matriciniai keitikliai neturėjo praktinio taikymo galimybių. Šiuo metu, dėl postūmio IGBT elementų gamyboje matriciniai keitikliai perdedami taikyti praktikoje.

Nors matriciniai dažnio keitikliai turi nemažai privalumų lyginant su tradiciniais dažnio keitikliais su tarpine nuolatinės įtampos grandimi, tačiau reikia paminėti, kad taip pat turi ir trūkumų. Šie dažnio keitikliai ypač jautrūs tiekiamos elektros energijos kokybei, kadangi jie neturi tarpinės nuolatinės įtampos grandies kuri tradiciniuose dažnio keitikliuose amortizuodavo trikdžius, todėl bet kokie įtampos šuoliai ar kritimai keitiklio įėjimo įtampoje, iš karto atsispindi ir išėjimo įtampoje, tai blogina variklio, maitinamo dažnio keitikliu, darbą [3].

Norint sumažinti įtampos šuolių ir kritimų įtaką tiesioginio matricinio dažnio keitiklio išėjimo įtampai, reikia naudoti modifikuotą elektroninių jungiklių valdymą. Modifikuotas valdymas buvo pasiūlytas studijose [4], [5], [6], tačiau jie rėmėsi nuostatos signalo arba moduliacijos indekso valdymo sistemoje padidiniu ir buvo tinkami kompensuoti įtampos kritimus tik iki 10% vardinės

įtampos. Didesniems įtampos trikdžiams, kurie pasitaiko Lietuvos energetinėje sistemoje šie metodai netaikytini.

Darbo tikslas – ištirti tiesioginio matricinio keitiklio galimybes kompensuoti įtampos trikdžius išplečiant erdviųjų vektorių formavimo segmentą ir pašalinant sumažėjusias įtampas iš erdvinio vektoriaus formavimo proceso.

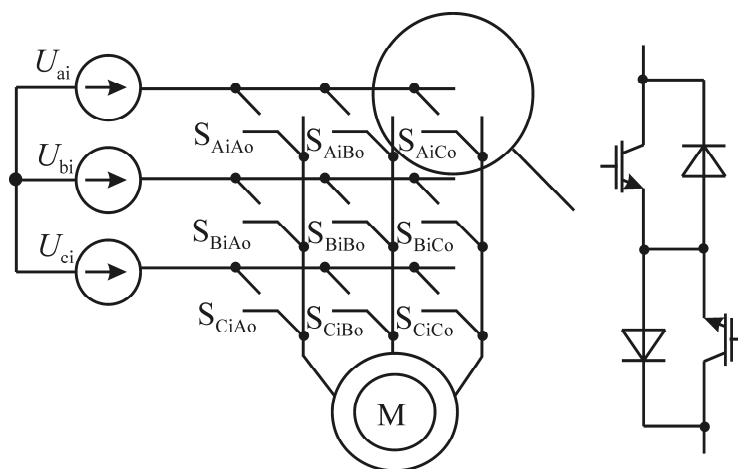
Darbo uždaviniai:

1. sudaryti Matlab Simulink modelį imituojantį 0,4kV elektros tinklo įtampas statistiškai pasitaikančių trikdžių atveju.
2. sudaryti matricinio keitiklio Matlab Simulink modelį su tradicine ir modifikuota erdvinio vektoriaus moduliacijos sistema.
3. atlikti eksperimentinį bandymą su fiziniu modeliu ir patikrinti Simulink modelio adekvatumą.
4. modeliavimo būdu ištirti modifikuotos erdvinio vektoriaus moduliacijos įtaką matricinio keitiklio išėjimo įtampai trikdžio metu.

1. Literatūros apžvalga

1.1. Tiesioginis matricinis dažnio keitiklis

Tiesioginis matricinis keitiklis, tai elektros energijos parametru (amplitudės ir dažnio) keitiklis, tiesiogiai prijungiantis apkrovą prie kintamosios įtampos šaltinio. Jis sudarytas iš $m \times n$ dvikrypčių jungikliu matricos, kurie prijungia m fazių įėjimo įtampos šaltinį prie n fazių apkrovos. Dažniausiai pasitaikanti matricinio keitiklio kombinacija 3×3 . Tokio tipo trifazis tiesioginis matricinis dažnio keitiklis (TMDK) yra sudarytas iš 9 dvikrypčių elektroninių jungiklių (dažniausiai IGBT pagrindu), kurie pagal valdymo algoritmą formuoja komutaciją tarp įėjimo ir išėjimo įtampų [2] (1 pav.).



1 pav. Trifazio tiesioginio matricinio dažnio keitiklio schema.

Lyginant su tradiciniu dažnio keitikliu matriciniai turi šiuos privalumus:

- Dvikryptis energijos srautas – matricinis keitiklis gali veikti kaip energijos iš elektros tinklo imtuvas ir energijos šaltinis (pvz. rekuperacinio stabdymo režime veikiančio variklio energijos gražinimui į tinklą);
- Nenaudojami energijos kaupimo elementai su ribota eksploatacijos trukme;
- Didesnis komunikacinių elementų kiekis leidžia tolygiau paskirstyti komunikacijos nuostolius;
- Įvairių valdymo strategijų pritaikymo galimybė leidžia sumažinti energijos nuostolius[11,12,13].

Lyginant su tradiciniu dažnio keitikliu matricinis keitiklis turi 2,5 karto didesnę galios ir dydžio santykį bei 5 kartus didesnę galios ir masės santykį [14].

Matriciniame keitiklyje, naudojant devynis dvikrypčius elektroninius jungiklius, teoriškai galima suformuoti 512 (2^9), skirtingas jungimo būsenų kombinacijas. Norint išvengti kritinių režimų, visų devynių dvikrypčių elektroninių jungiklių komutacija turi būti vykdoma taip, kad niekada nesusidarytų trumpas jungimas keitiklio įėjime, t.y. niekada nejungti tuo pačiu laiko momentu dviejų įėjimo fazių prie vienos išėjimo fazės. Esant apkrovoje induktyviajai dedamajai (kas dažniausiai pasitaiko jungiant elektros variklius), jungiklių komutacija turi būti vykdoma taip, kad nesusidarytų ir tuščioji veika (atvira grandinė) keitiklio išėjime. Įvertinus šias sąlygas naudotinas jungimo būsenų kombinacijų kiekis yra 27. [2].

Teoriškai galimos keletas TMDK elektroninių jungiklių komutacijos strategijų, tačiau dažniausiai naudojamas yra erdviųjų vektorių moduliacijos metodas [2]. Šis metodas ir bus naudojamas šiame darbe.

Išėjimo ir įėjimo įtampų matricos:

$$\mathbf{u}_o = \begin{pmatrix} \mathbf{u}_{Ao} \\ \mathbf{u}_{Bo} \\ \mathbf{u}_{Co} \end{pmatrix} \quad \mathbf{u}_i = \begin{pmatrix} \mathbf{u}_{Ai} \\ \mathbf{u}_{Bi} \\ \mathbf{u}_{Ci} \end{pmatrix}$$

Priklausomybė tarp išėjimo ir įėjimo įtampų, bei srovių nuo dvikrypčių jungiklių būsenos, gali būti aprašoma matricos forma:

$$\begin{pmatrix} \mathbf{u}_{Ao} \\ \mathbf{u}_{Bo} \\ \mathbf{u}_{Co} \end{pmatrix} = \begin{pmatrix} S_{AiAo}(t) S_{BiAo}(t) S_{CiAo}(t) \\ S_{AiBo}(t) S_{BiBo}(t) S_{CiBo}(t) \\ S_{AiCo}(t) S_{BiCo}(t) S_{CiCo}(t) \end{pmatrix} \begin{pmatrix} \mathbf{u}_{Ai} \\ \mathbf{u}_{Bi} \\ \mathbf{u}_{Ci} \end{pmatrix}.$$

Perdavimo matrica:

$$T = \begin{pmatrix} S_{AiAo}(t) S_{BiAo}(t) S_{CiAo}(t) \\ S_{AiBo}(t) S_{BiBo}(t) S_{CiBo}(t) \\ S_{AiCo}(t) S_{BiCo}(t) S_{CiCo}(t) \end{pmatrix}$$

$$\mathbf{u}_o = T \times \mathbf{u}_i$$

Ryšys tarp išėjimo ir įėjimo srovių:

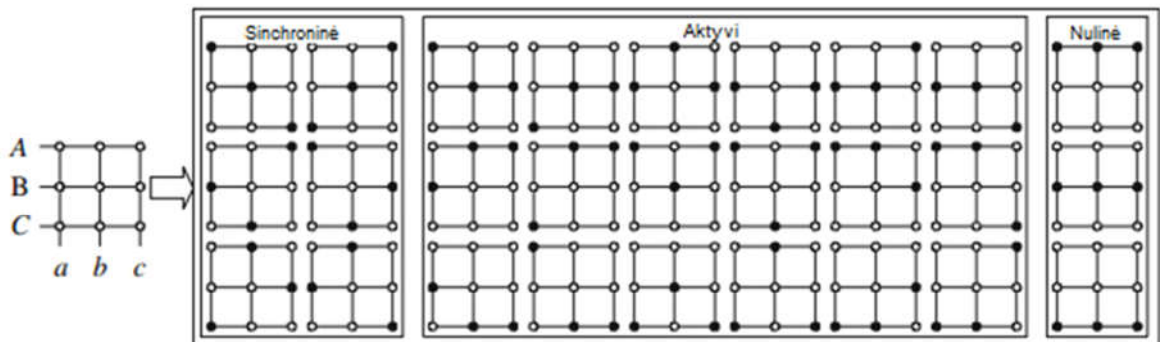
$$i_i = T^T \times i_o$$

T^T - transponuotoji matrica

Čia, u_o – išėjimo įtampa; u_s – įėjimo įtampa; $\alpha = e^{j(2\pi/3)}$.

27 galimos jungiklių kombinacijos yra skirstomos į tris grupes:

1. Nulinė – visos trys išėjimo fazės jungiamos prie vienos įėjimo fazės, taip išėjime gaunama nulinė įtampa ir srovė;
2. Synchroninė – visos išėjimo fazės jungiamos prie skirtingų įėjimo fazių. Šiuo atveju išėjimo įtampos ir srovės vektoriai turi kintamas kryptis ir negali būti naudojami generuojant išėjimo įtampą.
3. Aktyvi – 18 kombinacijų, kuriose išėjimo įtampos ir įėjimo srovės vektoriai turi fiksuotas kryptis. Šių vektorių dydis priklauso nuo momentinės įėjimo linijinės įtampos ir išėjimo fazių srovės. Šiuo atveju dvi išėjimo fazės yra sujungiamos su viena įėjimo faze [2].

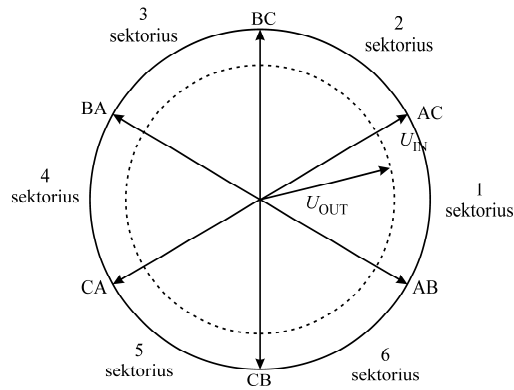
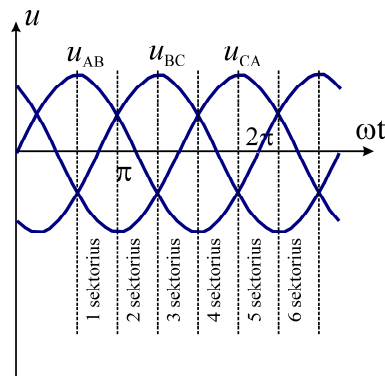


2 pav. Jungiklių kombinacijos [2].

Naudojant erdviųjų vektorių moduliacijos metodą, 2 jungiklių grupė nenaudojama, išėjimas generuojamas iš 1 ir 3 grupės. Antros grupės jungiklių konfigūracijos pagal 1 lentelę yra suskirstomos į sektorius (3 pav.), pagal tai, kurių fazių linijinė įtampa tuo metu yra nulinė.

1 lentelė. Išėjimo įtampos bei įėjimo srovės priklausomybė nuo įjungtų elektroninių jungiklių konfigūracijos [2].

No.	Vector No.	a	b	c	S_{aA}	S_{aB}	S_{aC}	S_{bA}	S_{bB}	S_{bC}	S_{cA}	S_{cB}	S_{cC}	u_{ab}	u_{bc}	u_{ca}	i_A	i_B	i_C
1	0_A	A	A	A	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
2	0_B	B	B	B	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0
3	0_C	C	C	C	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0
4	-3	A	C	C	1	0	0	0	0	1	0	0	1	$-u_{CA}$	0	u_{CA}	i_a	0	$-i_a$
5	+2	B	C	C	0	1	0	0	0	1	0	0	1	u_{BC}	0	$-u_{BC}$	0	i_a	$-i_a$
6	-1	B	A	A	0	1	0	1	0	0	1	0	0	$-u_{AB}$	0	u_{AB}	$-i_a$	i_a	0
7	+3	C	A	A	0	0	1	1	0	0	1	0	0	u_{CA}	0	$-u_{CA}$	$-i_a$	0	i_a
8	-2	C	B	B	0	0	1	0	1	0	0	1	0	$-u_{BC}$	0	u_{BC}	0	$-i_a$	i_a
9	+1	A	B	B	1	0	0	0	1	0	0	1	0	u_{AB}	0	$-u_{AB}$	i_a	$-i_a$	0
10	-6	C	A	C	0	0	1	1	0	0	0	0	1	u_{CA}	$-u_{CA}$	0	i_b	0	$-i_b$
11	+5	C	B	C	0	0	1	0	1	0	0	0	1	$-u_{BC}$	u_{BC}	0	0	i_b	$-i_b$
12	-4	A	B	A	1	0	0	0	1	0	1	0	0	u_{BA}	$-u_{BA}$	0	$-i_b$	i_b	0
13	+6	A	C	A	1	0	0	0	0	1	1	0	0	$-u_{CA}$	u_{CA}	0	$-i_b$	0	i_b
14	-5	B	C	B	0	1	0	0	0	1	0	1	0	u_{BC}	$-u_{BC}$	0	0	$-i_b$	i_b
15	+4	B	A	B	0	1	0	1	0	0	0	1	0	$-u_{AB}$	u_{AB}	0	i_b	$-i_b$	0
16	-9	C	C	A	0	0	1	0	0	1	1	0	0	0	u_{CA}	$-u_{CA}$	i_c	0	$-i_c$
17	+8	C	C	B	0	0	1	0	0	1	0	1	0	0	$-u_{BC}$	u_{BC}	0	i_c	$-i_c$
18	-7	A	A	B	1	0	0	1	0	0	0	1	0	0	u_{AB}	$-u_{AB}$	$-i_c$	i_c	0
19	+9	A	A	C	1	0	0	1	0	0	0	0	1	0	$-u_{CA}$	u_{CA}	$-i_c$	0	i_c
20	-8	B	B	C	0	1	0	0	1	0	0	0	1	0	u_{BC}	$-u_{BC}$	0	$-i_c$	i_c
21	+7	B	B	A	0	1	0	0	1	0	1	0	0	0	$-u_{AB}$	u_{AB}	i_c	$-i_c$	0
22	-	A	B	C	1	0	0	0	1	0	0	0	1	u_{AB}	u_{BC}	u_{CA}	i_a	i_b	i_c
23	-	A	C	B	1	0	0	0	0	1	0	1	0	$-u_{CA}$	$-u_{BC}$	$-u_{AB}$	i_a	i_c	i_b
24	-	B	A	C	0	1	0	1	0	0	0	0	1	$-u_{AB}$	$-u_{CA}$	$-u_{BC}$	i_b	i_a	i_c
25	-	B	C	A	0	1	0	0	0	1	1	0	0	u_{BC}	u_{CA}	u_{AB}	i_c	i_a	i_b
26	-	C	A	B	0	0	1	1	0	0	0	1	0	u_{CA}	u_{AB}	u_{BC}	i_b	i_c	i_a
27	-	C	B	A	0	0	1	0	1	0	1	0	0	$-u_{BC}$	$-u_{AB}$	$-u_{CA}$	i_c	i_b	i_a



3 pav. Erdvinio vektoriaus formavimas sektoriuose [2]

Reikiamas išėjimo įtampos vektorius U_0 formuojamas iš dviejų vienodo ilgio vektorių apskaičiuojant kiekvieno vektoriaus (su vektoriumi susijusios jungiklių kombinacijos) išlaikymo trukmę. Kaitaliojant jungiklių kombinaciją ir išlaikymo trukmę formuojama reikiamo dažnio, amplitudės ir fazės kampo įtampa. Erdvinio vektoriaus padėties sektorius parenkamas priklausomai nuo linijinių įtampų momentinių verčių. Pavyzdžiui, vektoriaus padėties 1 sektoriuje laiko intervale didžiausias skirtumas tarp AB ir AC linijinių įtampų. Pasisukus 60° didžiausias skirtumas bus tarp BC ir AC linijinių įtampų. Todėl šios įtampos ir naudojamos vektoriaus pozicijai 2 sektoriuje formuoti.

$$U_{AB0} = U_{ABi}(d_{\alpha\mu} + d_{\beta\mu}) + U_{ACi}(d_{\alpha\vartheta} + d_{\beta\vartheta})$$

$$U_{BC0} = U_{ABi}(-d_{\alpha\mu}) + U_{ACi}(-d_{\alpha\vartheta})$$

$$U_{CA0} = U_{ACi}(-d_{\beta\mu}) + U_{ACi}(-d_{\beta\vartheta})$$

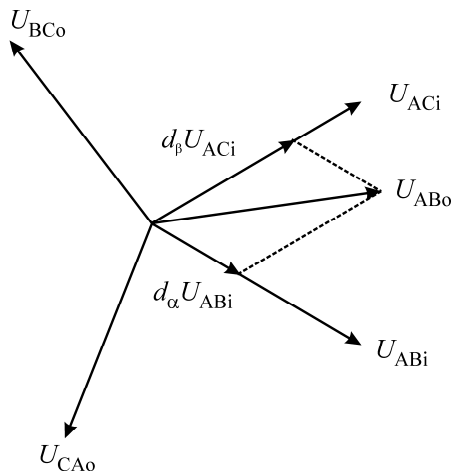
$$d_{\alpha\mu} = d_\alpha d_\mu = m \sin(60^\circ - \theta_{SV}) \sin(60^\circ - \theta_{SC})$$

$$d_{\beta\mu} = d_\beta d_\mu = m \sin(\theta_{SV}) \sin(60^\circ - \theta_{SC})$$

$$d_{\alpha\vartheta} = d_\alpha d_\vartheta = m \sin(60^\circ - \theta_{SV}) \sin(\theta_{SC})$$

$$d_{\beta\vartheta} = d_\beta d_\vartheta = m \sin(\theta_{SV}) \sin(\theta_{SC})$$

Čia: θ_{SV} invertavimo kampas susietas su išėjimo įtampos dažniu, kinta $0-60^\circ$ sektoriaus viduje. θ_{SC} lyginimo kampas, susietas su tinklo įtampos dažniu. Išėjimo linijinė įtampa U_{AB0} generuojama iš dviejų linijinių įėjimo įtampų U_{ABi} ir U_{ACi} . Likusios Linijinės išėjimo įtampos generuojamos iš vienos likusios įėjimo įtampos.



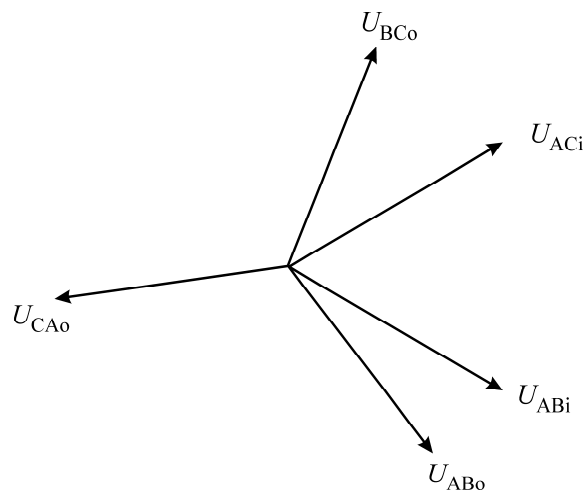
4 pav. Išėjimo įtampos formavimas 1 sektoriuje

Pasisukus išėjimo įtampos vektoriui 60° kampui prasideda sekantis sektorius. Tuomet maksimalią reikšmę įgauna linijinės įtampos U_{CA} vektorius. Vektorių padėtys šiuo atveju pateiktos 5 pav. Lygtys šiame sektoriuje:

$$U_{CAo} = -(d_{\alpha\mu} + d_{\beta\mu})(-U_{ABi}) + (d_{\alpha\vartheta} + d_{\beta\vartheta})(-U_{ACi})$$

$$U_{ABo} = U_{ABi}(-d_{\alpha\mu}) + U_{ACi}(-d_{\alpha\vartheta})$$

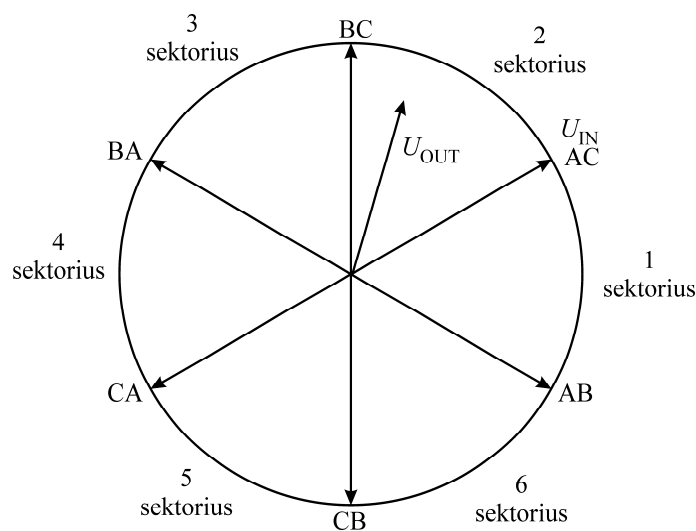
$$U_{BCo} = (-U_{ACi})(-d_{\beta\mu}) + (-U_{ACi})(-d_{\beta\vartheta})$$



5 pav. Išėjimo įtampos formavimas pasikeitus maksimalaus skirtumo įtampoms

Tokiu būdu besisukant U_{AB0} , U_{BC0} , U_{CA0} vektoriams ir besikeičiant sektoriams kai θ_{SV} padidėja iki 60° formuojama trifazė išėjimo įtampa.

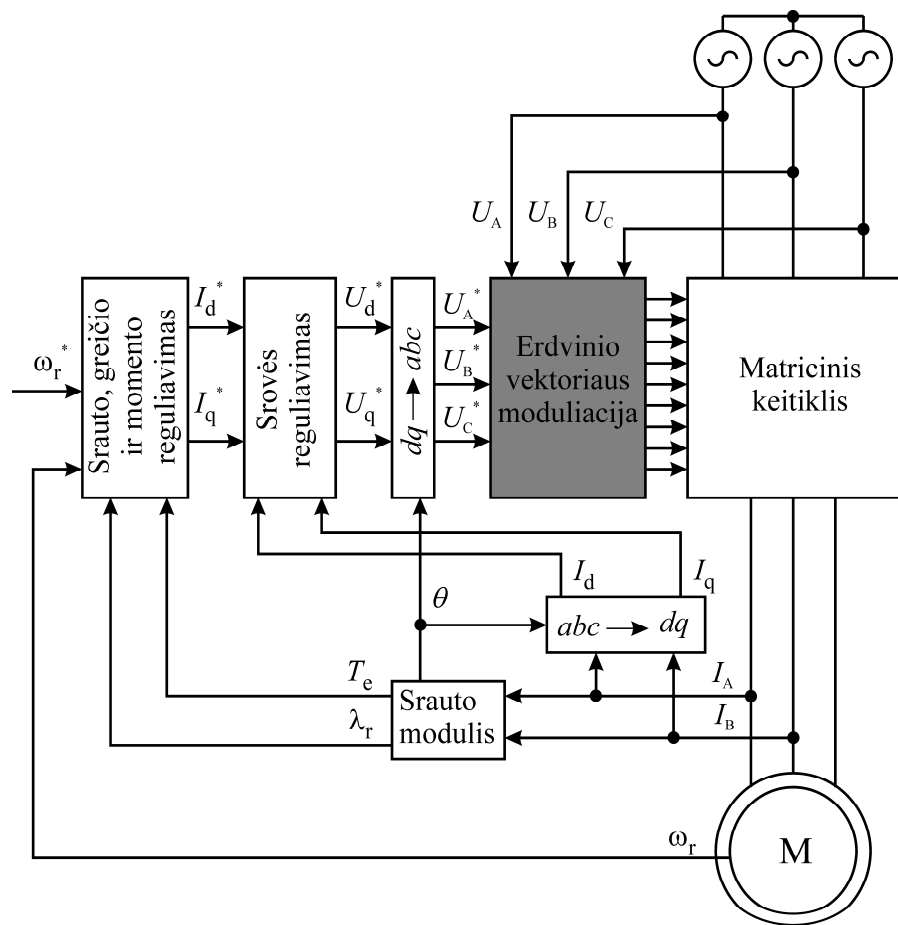
Kintant tinklo įtampos fazės kampui kaičiasi ir linijinės įtampos, tarp kurių yra didžiausias skirtumas, tuo pačiu keičiasi lyginimo stadijos sektorius. Pasikeitus sektoriui (iš 1 į 2) įtampų vektoriai pateikti 6 pav. Maksimalus vektorius formuojamas iš U_{ACi} ir U_{BCi} vektorių.



6 pav. Erdvinio vektoriaus formavimas 2 sektoriuje

Tokiu būdu įėjimo sektorių nulemia kampas θ_{SC} , jis sinchronizuojamas su tinklo dažniu ir per tinklo periodą šešis kartus kinta intervale $0 \div 60^\circ$. Išėjimo sektorių nulemia kampas θ_{SV} , kampo kitimo greitis nulemia keitiklio išėjimo įtampos dažnį f_0 .

Uždara matricinės pavaros valdymo schema pateikta 7 paveiksle[16].



7 pav. Matricinės pavaros valdymo sistema

Šioje sistemoje panaudoti „Clarke“ ir „Park“ transformacijos moduliai. „Clarke“ transformacija leidžia trijų fazinių srovių $i_A(t)$, $i_B(t)$, ir $i_C(t)$ sukurtą srovės vektorių išreikšti stačiakampėje koordinatinių sistemoje srovių i_α ir i_β komponentėmis. „Park“ transformuoja dvifazę stačiakampę koordinatinių sistemą į besisukančią d, q koordinatinių sistemą. Srauto modulis pagal dviejų fazių srovių dedamąsias formuoja grįžtamo ryšio signalus pagal momentą ir posūkio kampą. Pagal šiuos signalus ir grįžtamojo ryšio pagal greitį signalą srauto, greičio ir momento reguliavimo sistemoje apskaičiuojami keitiklio srovės nuostatos signalai d, q koordinatinių sistemoje. Srovės reguliavimo sistemoje apskaičiuojamos nuostatos signalo pagal įtampą dedamosios d, q koordinatinių sistemoje, kurios transformuojamos į fazines dedamąsias ir erdvinio vektoriaus moduliacijos blokui. Pagal nuostatos įtampas U_A^* , U_B^* , U_C^* erdvinio vektoriaus moduliacijos blokas formuoja matricinio keitiklio dvikrypčių elektroninių jungiklių valdymo signalus.

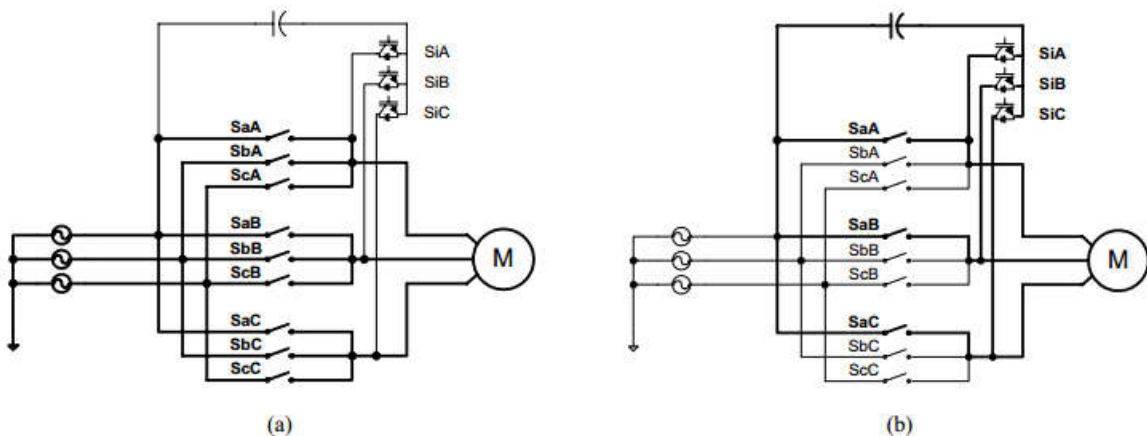
1.2. Tiesioginio matricinio dažnio keitiklio valdymas veikiant maitinimo įtampos trikdžiams

Teoriškai įtampos trikdžių kompensavimas gali būti kelių rūšių [7]:

1. Energijos išlaikymas – trikdžių kompensavimas, kai variklis turi veikti išlaikydamas savo greitį ir sukimo momentą nepaisant elektros tiekimo sutrikimų panaudojant reaktyviuosiuose elementuose sukauptą energiją. Energijos išlaikymo matriciniuose dažnio keitikliuose neįmanomas realizuoti, nes šis keitiklis neturi reaktyviųjų elementų.
2. Inercijos išlaikymas – šis metodas taikomas, kai variklio apkrova turi didelę masę ir inerciją, toks išlaikymas realizuojamas, kai keitikliui leidžiama valdyti regeneruojamą energiją, tai reiškia, kad variklis elektros sutrikimo metu dirba generatoriaus režime.
3. Baterijos naudojamos tada, kai reikia užtikrinti ilgą darbo laiką, nutrūkus elektros energijos tiekimui.
4. Atsarginės inercijos išlaikymas – šiam būdai naudojamas papildomas didelės masės smagratės, o principas toks pats kaip ir antrojo būdo.

1.2.1 Įtampos trikdžių kompensavimas panaudojant papildomą energijos kaupiklį

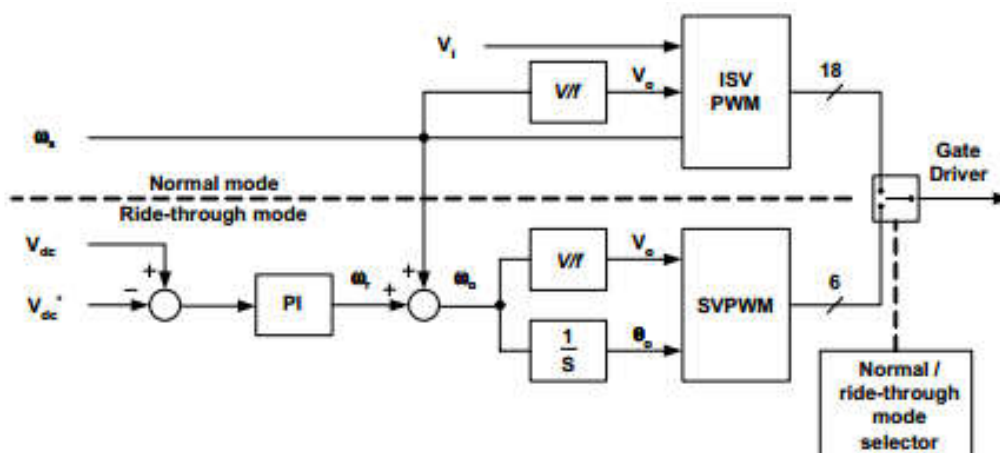
Literatūroje [8] darbe siūlomas įtampos kritimo kompensavimo būdas į matricinio keitiklio schemą įtraukia papildomus tris jungiklius ir vieną nuolatinės įtampos grandies kondensatorių. Dingus elektros įtampai tinkle, išjungiami 6 iš 9 keitiklio jungiklių ir matricinis dažnio keitiklis tampa įprastiniu keitikliu su nuolatinės įtampos grandimi (8 pav.)



8 pav. a) normalus TMDK veikimas b) dažnio keitiklio veikimas esant elektros tiekimo sutrikimams

Kadangi kondensatoriuje sukauptos energijos paprastai neužtenka variklio maitinimui, todėl variklio sukauptą inerciją naudojama nuolatinės įtampos grandies įtampai palaikyti. Įtampai

palaikyti naudojamas PI reguliatorius, 9 paveiksle pavaizduota pasiūlytos sistemos blokinė schema [8].



9 pav. Pasiūlyto sprendimo blokinė schema

Tokio pasiūlyto metodo privalumai yra šie:

1. Įtampos sutrikimų metu išlaikoma sinchronizacija tarp variklio ir dažnio keitiklio;
2. Variklis gali pasiekti prarastą greitį be srovės perėjimų, kai atstatoma elektros įtampa tinkle.

Tačiau norint įgyvendinti šį pasiūlymą keitiklio sandaroje atsiranda papildomų elementų, padidinančių keitiklio kainą ir sumažinančių patikimumą.

1.2.2 Įtampos trikdžių kompensavimas didinant moduliacijos koeficientą

Šis metodas pasiūlytas [9] darbe nereikalauja jokių papildomų elementų. Kompensavimas vyksta tik modifikuojant algoritmą. Elektros tinkle įvykus gedimui, dėl kurio nukrenta bet kurios fazės maitinimo įtampa, nukrenta ir virtualios nuolatinės įtampos grandies įtampa matriciniame dažnio keitiklyje.

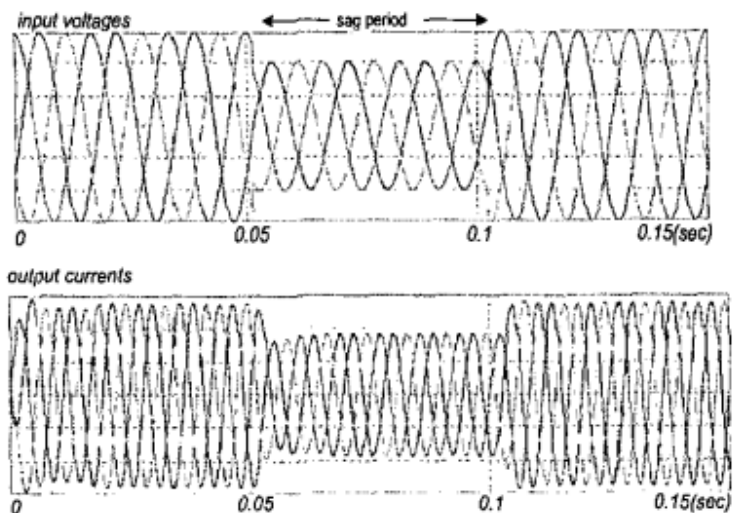
Moduliacijos koeficientas:

$$m_{rec} = \frac{2U_{DC}}{3U_{im}};$$

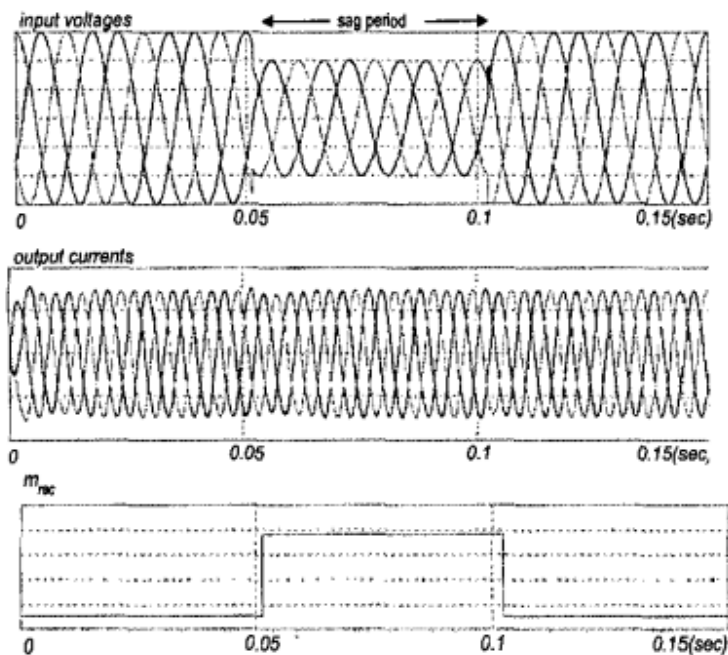
čia, m_{rec} – moduliacijos koeficientas, U_{DC} – virtualios nuolatinės įtampos grandies koeficientas; U_{im} – keitiklio įėjimo įtampa.

Nukritus arba padidėjus maitinimo įtampai, išlaikant tą patį moduliacijos koeficientą kinta ir išėjimo įtampa. Norint išlaikyti tą pačią išėjimo įtampą, siūloma keisti moduliacijos koeficientą. 10

ir 11 paveiksluose pavaizduota kaip kinta išėjimo įtampa naudojant algoritmą be modifikacijos ir su modifikacija [9].



10 pav. Įėjimo įtampos ir išėjimo įtampos laiko diagramos išlaikant nekintantį moduliacijos koeficientą



11 pav. Įėjimo įtampos ir išėjimo įtampos laiko diagramos kompensuojant trikdžius paaukštintu moduliacijos koeficientu

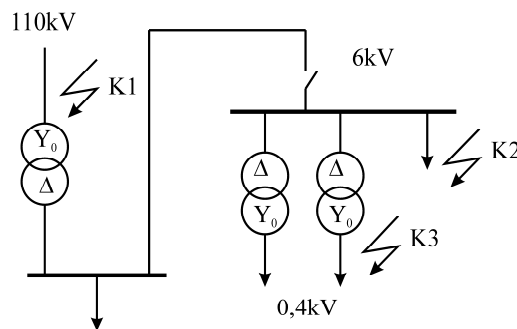
Kaip matyti iš 10 ir 11 paveikslų pasiūlytas algoritmas kompensuoja įtampos kritimą. Tokio sprendimo didžiausias privalumas yra tas, kad nereikia jokių papildomų elementų. Toks maitinimo įtampos trikdžių kompensavimo būdas netinka tokiu atveju, kai trikdžio metu pakinta įtampų fazės kampai arba įtampa kurioje nors fazėje sumažėja iki nulio.

1.2.3 Įtampos kompensavimas didinant įtampos nuostatos signalą

Literatūros šaltinyje [17] pasiūlytas trikdžių įtakos kompensavimo būdas valdymo sistemoje padidinantis įtampos nuostatos signalą. Tačiau, įtampų nesimetriškumą sukeliančių trikdžių atveju šis padidinimas sukels dar didesnę išėjimo įtampų nesimetriškumą.

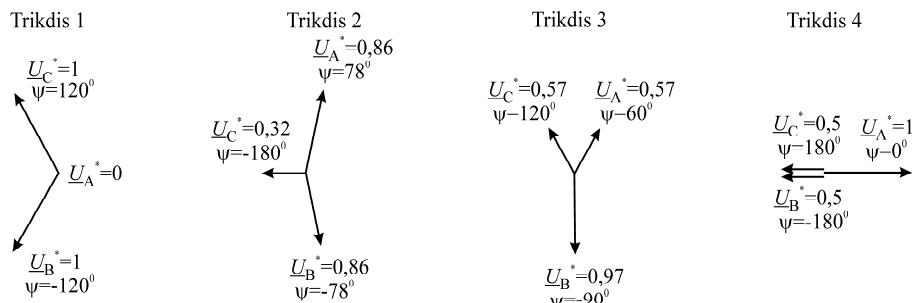
1.3. Įtampos trikdžiai Lietuvos elektros energijos tiekimo linijose

Remiantis statistiniais duomenimis [15] įtampos trikdžiai 0,4kV elektros energijos tiekimo linijose pasireiškia dėl gedimų energetinės sistemos taškuose K1, K2 ir K3 (pav. 12).



12 pav. Gedimai energetinėje sistemoje sukeltys trikdžius 0,4kV tinkle

Minėti gedimai sukelia įtampos 0,4kV tinkle pokyčius, kuriuos vaizduoja 13 pav. vektorinės diagramos.

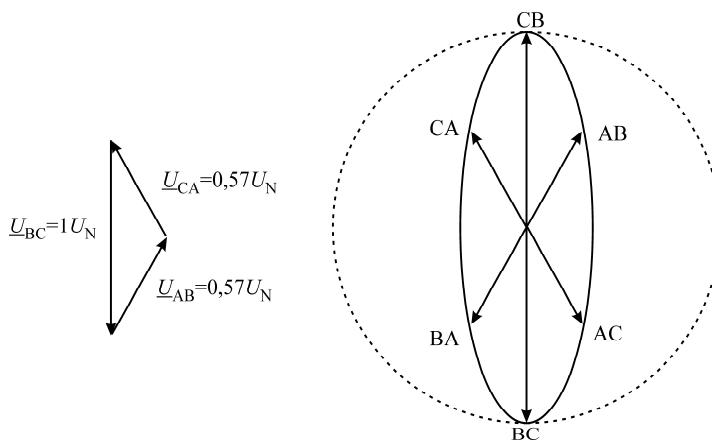


13 pav. Trikdžių sukelti įtampos 0,4kV tinkle pokyčiai

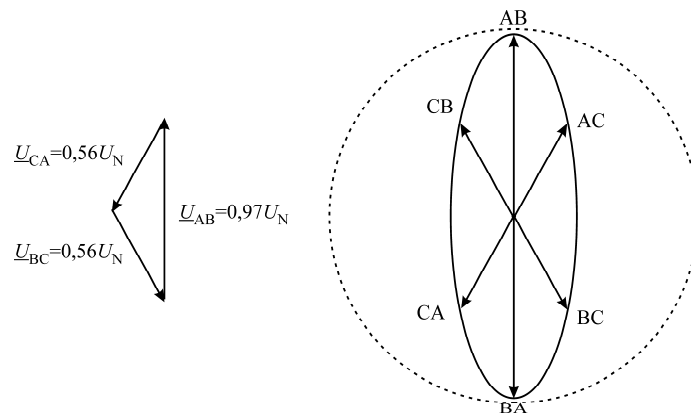
Šie pokyčiai įvardinti kaip keturių tipų trikdžiai, kurių metu įvyksta įtampos amplitudės sumažėjimas ir fazės kampo poslinkis. Trikdžio 1 atveju fazinė įtampa fazėje A sumažėja iki nulio, trikdžio 2 atveju įtampa fazėje C sumažėja iki 0,32 vardinės įtampos. Trikdžio 4 atveju fazinių įtampų kampai B ir C fazėse susilygina.

2. Erdvinio vektoriaus moduliacija trikdžių atveju

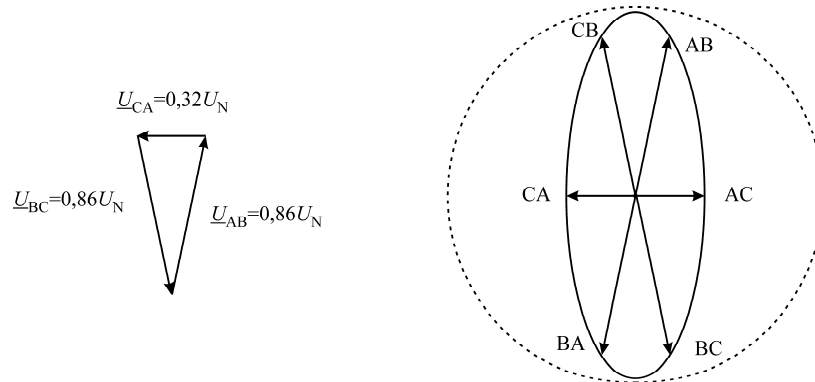
Įtampos trikdžių energetinės sistemos elektros tiekimo linijose atveju matricinio keitiklio išėjimo įtampos formavimas tampa komplikuoatas. Skirtingai negu tradicinis dažnio keitiklis su energiją kaupiančiais ir įtampos sumažėjimą kompensuojančiais elementais nuolatinės srovės grandyje, matriciniame keitiklyje ženkliai pasikeičia erdvinės moduliacijos vektorių amplitudė ir kampas (pav. 14-17). Pavyzdžiui, trikdžio 3 atveju (pav.15), linijinės įtampos vektorius tarp fazių A ir C sutrumpėja iki 32% vardinės įtampos. Trikdžio 4 atveju, erdvinio vektoriaus formavime dalyvauja tik dvi linijinės įtampos (pav. 4). Trikdžio 1 atveju linijinės įtampos vektoriaus amplitudė sumažėja iki 57% vardinės, o trikdžio 2 atveju iki 56% vardinės. Visa tai lemia erdvinio vektoriaus moduliacijos sekstanto formos pasikeitimą (pav. 14-17).



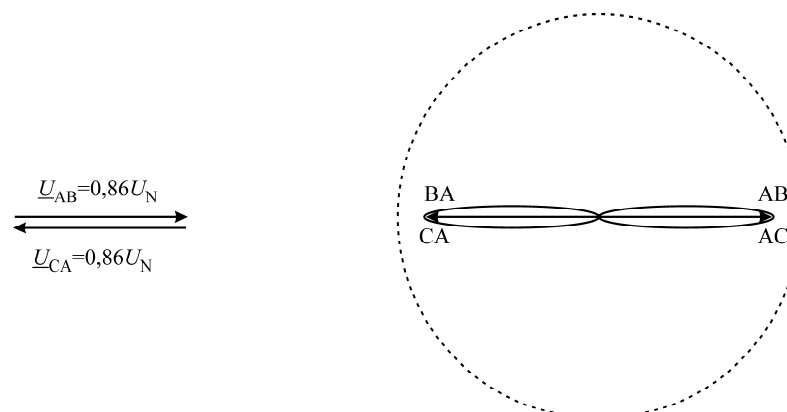
Pav. 14. Linijinių įtampų vektorinė diagrama ir erdvinio vektoriaus moduliacijos zona trikdžio 1 atveju



Pav. 15. Linijinių įtampų vektorinė diagrama ir erdvinio vektoriaus moduliacijos zona trikdžio 2 atveju



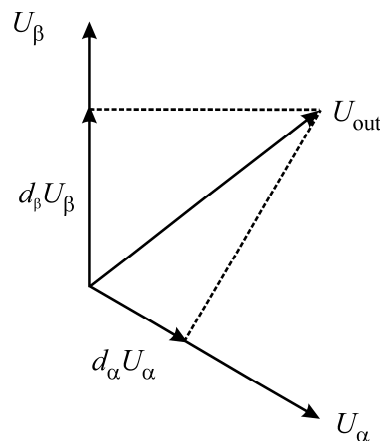
Pav. 16. Linijinių įtampų vektorinė diagrama ir erdvinio vektoriaus moduliacijos zona trikdžio 3 atveju



Pav. 17. Linijinių įtampų vektorinė diagrama ir erdvinio vektoriaus moduliacijos zona trikdžio 4 atveju

Be linijinių įtampų vektorių amplitudės pasikeitimų įvyksta ir ženklus fazės kampų poslinkis. Pavyzdžiui trikdžio 1 atveju vektoriaus \underline{U}_{BC} , o trikdžio 2 atveju vektoriaus \underline{U}_{AB} fazės kampo nuokrypis 30° (pav. 14, 15). Trikdžio 4 atveju fazės kampo nuokrypis tarp vektorių \underline{U}_{AB} ir \underline{U}_{CA} sudaro 60° .

Bendras linijinių įtampų pokyčio požymis visų trikdžių atveju - esminis pokytis įvyksta vienoje arba dviejose fazėse. Reiškia išlieka galimybė modifikuojant erdvinio vektoriaus moduliacijos sektorių suformuoti pakankamos variklio darbui amplitudės įtampą matricinio keitiklio išėjime iš mažiausiai iškreiptų įėjimo įtampų. Tai susivestų į sumažėjusios įtampos kompensavimą likusių fazių įtampomis. Tai įmanoma padaryti trikdžio atveju išplečiant sektoriaus kampą nuo 60° iki 120° , tuo pačiu pašalinant sumažėjusias dėl trikdžio įtampas iš erdvinio vektoriaus formavimo sekos (pav. 18).



Pav. 18. Išėjimo įtampos formavimas trikdžio atveju

Teoriškai toks sektoriaus išplėtimas taikytinas tik trikdžio atveju, normalaus energijos tiekimo režime išplėtimas sumažintų išėjimo įtampą.

Įvertinant elektros energijos tiekimo linijų įtampos lygį ir vykdant tradicinę EVM normalaus energijos tiekimo režime arba sumažėjusios įtampos kompensavimo EVM trikdžio atveju, vektoriaus dedamųjų skaičiavimas vykdomas pagal tokią seką:

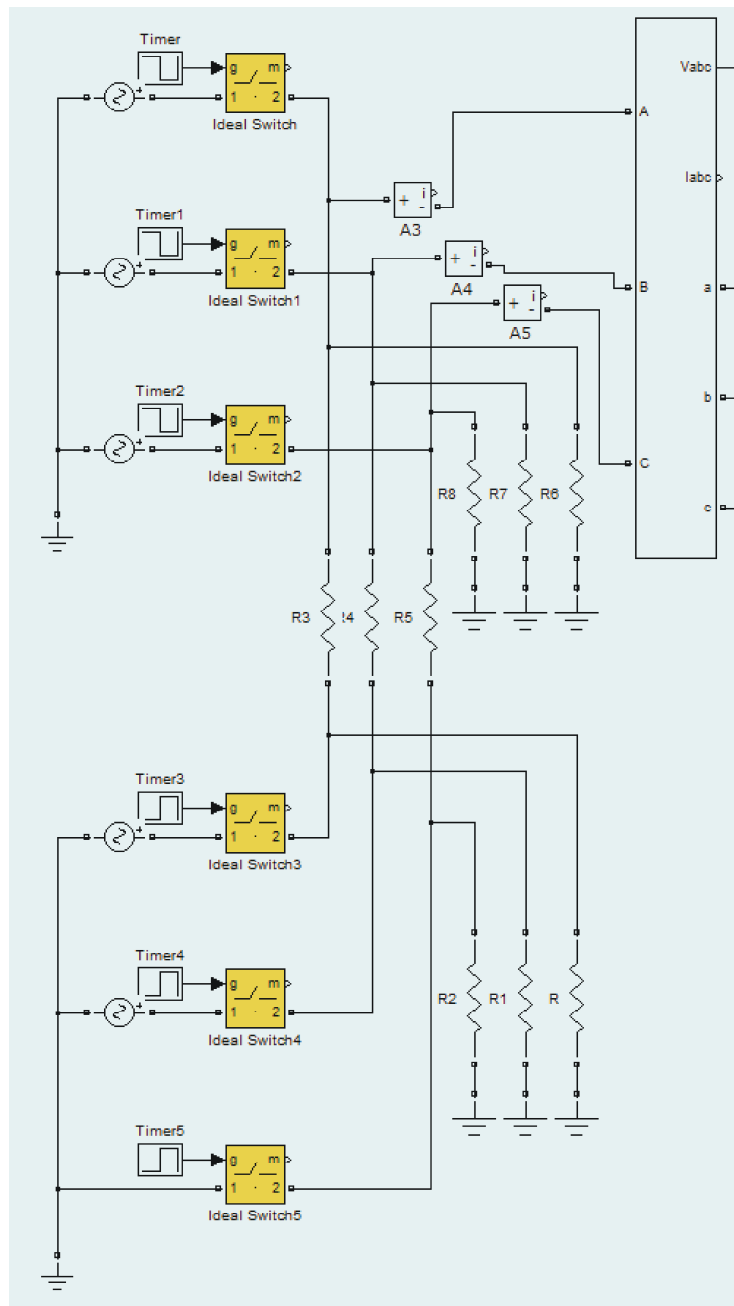
1. Pradinių duomenų θ_{out} , f_{out} , f_{sw} ; reikalingų erdvinio vektoriaus moduliacijai, nuskaitymas;

2. Įtampos U_{in} energijos tiekimo linijose lygio įvertinimas lyginant su ribine verte ir modifikuotos EVM (kampo išplėtimas, sumažėjusių įtampų šalinimas iš erdvinio vektoriaus formavimo proceso) įjungimas, jei įtampa mažesnė už ribinę vertę;
3. Sektoriaus vektorių sekstante nustatymas pagal įtampos fazės kampą θ_{out} normalaus energijos tiekimo režime. Jei kampas ribose $-30^\circ < \theta_{out} < +30^\circ$, $S = 1$, jei kampas ribose $30^\circ < \theta_{out} < 90^\circ$, $S = 2$, jei kampas ribose $90^\circ < \theta_{out} < 150^\circ$, $S = 3$ ir t.t.;
4. Sektoriaus vektorių sekstante nustatymas pagal įtampos fazės kampą θ_{out} trikdžio atveju. Jei kampas ribose $0^\circ < \theta_{out} < 60^\circ$, $S = 1$, jei kampas ribose $60^\circ < \theta_{out} < 120^\circ$, $S = 2$, jei kampas ribose $120^\circ < \theta_{out} < 180^\circ$, $S = 3$ ir t.t.;
5. Komutacijos kampo θ_{sw} sektoriaus S viduje skaičiavimas remiantis θ_{out} , sektoriui $S = 1$, $\theta_{sw} = \theta_{out} - (-30^\circ)$; sektoriui $S = 2$, $\theta_{sw} = \theta_{out} - 30^\circ$ ir t.t. normalaus energijos tiekimo režime;
6. Komutacijos kampo θ_{sw} sektoriaus S viduje skaičiavimas remiantis θ_{out} , sektoriui $S = 1$, $\theta_{sw} = \theta_{out}$; sektoriui $S = 2$, $\theta_{sw} = \theta_{out} - 60^\circ$ ir t.t. trikdžio atveju;
7. Komutacijos dedamųjų d_α , d_β , d_0 skaičiavimas remiantis θ_{sw} ;
8. Komutacijos trukmės T_α , T_β , T_0 skaičiavimas remiantis $T_\alpha = d_\alpha \cdot T_{sw}$ ir t.t., $T_{sw} = 1/f_{sw}$.
9. Loginiai veiksmai dvikrypčių elektroninių jungiklių valdymo signalams formuoti, sektoriuje $S = 1$; signalai formuojami remiantis 60° kampu perstumtais įtampų U_{AC} ir U_{AB} vektoriais, sektoriuje $S = 2$, įtampų U_{AB} ir U_{CB} vektoriais, sektoriuje $S = 3$, įtampų U_{CB} ir U_{CA} vektoriais ir t.t. normalaus energijos tiekimo režime;
10. Loginiai veiksmai dvikrypčių elektroninių jungiklių valdymo signalams formuoti, sektoriuje $S = 1$; signalai formuojami remiantis 120° kampu perstumtais įtampų U_{AC} ir U_{AB} vektoriais, sektoriuje $S = 2$, įtampų U_{AB} ir U_{CB} vektoriais, sektoriuje $S = 3$, įtampų U_{CB} ir U_{CA} vektoriais ir t.t. trikdžio atveju;
11. Impulsų pločio moduliacijos formavimas remiantis T_α , T_β ir T_0 ;
12. Didinamas θ_{sw} ir pradama nuo punkto 2.

3. Matlab Simulink modelis

3.1. Trikdžius energetinėje sistemoje imituojantis modelis

Remiantis 13 pav. pateiktomis įtampų trikdžio atveju vektorinėmis diagramomis sudaryti Matlab Simulink modeliai visiems keturiems trikdžių tipams.



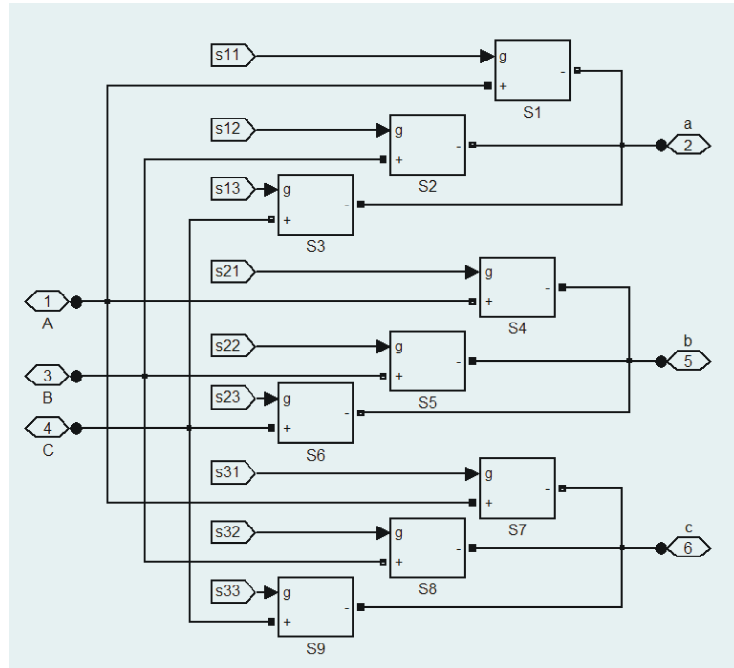
Pav. 19. Matlab Simulink modelio fragmentas trikdžiams 0,4kV tinkle imituoti

Paveiksle 19 patektame modelyje panaudoti bloka:

- „Ideal sinusoidal AC voltage“ tinklo įtampas normalaus darbo režime ir trikdžio metu imituojantys bloka;
- „Ideal switch“ – pagal taimerio signalą atjungiantys normalaus darbo įtampas imituojančius šaltinius ir prijungiantys įtampas trikdžio atveju imituojančius šaltinius.
- „Timer“ – taimeris duodantis signalą jungiklių persijungimui.
- „Three-phase VI measurement“ – įtampų ir sroviu matavimo blokas.

3.2. Matricinio keitiklio jėgos grandinių modelis

Matricinio keitiklio jėgos grandinių modelis (pav. 20) sudarytas iš devynių dvikrypčių elektroninių jungiklių S1-S4 kurių atstojamoji schema ir parametrai pateikti paveiksle 21.



Pav. 20. Matricinio keitiklio jėgos grandines imituojantis modelis

Pav. 21. Dvikrypčius elektrinius jungiklius imituojantis modelis

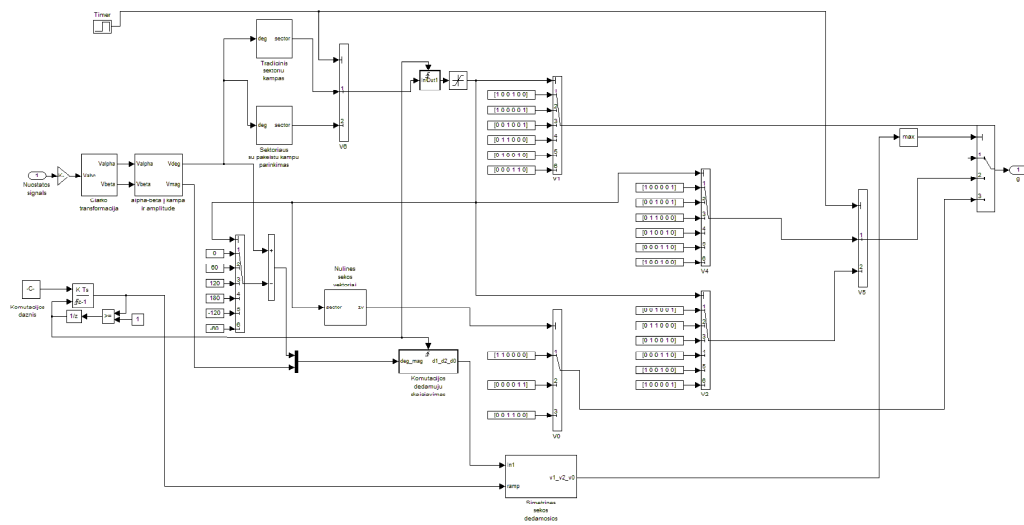
Jungiklių imitacijai panaudoti šie blokai:

- „IGBT mask“ – dvipolį tranzistorių su izoliuotą užtūrą imituojantis blokas;
- „Diode“ – diodą imituojantis blokas.

IGBT ir diodų parametrai palikti tokie, kokie Matlab Simulink siūlomi pagal nutylėjimą.

3.3. Erdvinio vektoriaus moduliacijos modelis

Erdvinio vektoriaus moduliacijos modelis pateiktas 22 paveiksle.

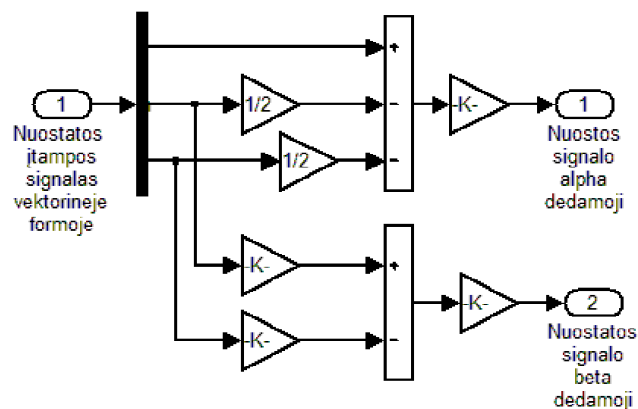


Pav. 22. Erdvinio vektoriaus moduliacijos signalą formuojanti posistemė.

Šis modelis sudarytas iš tokių Matlab Simulink blokų ir posistemų:

- „abc 2 alpha-beta“ - „Clarke“ A, B, C \Rightarrow α, β transformaciją atliekanti posistemė. Clarke transformacija leidžia trijų fazinių įtampų $u_A(t)$, $u_B(t)$, ir $u_C(t)$ sukurtą įtampų vektorių išreikšti stačiakampėje koordinatinių sistemoje srovių i_α ir i_β komponentėmis;
- „alpha-beta 2 deg-mag“ – stačiakampėje koordinatinių sistemoje išreikštą vektorių aprašo vektoriaus amplitudės ir kampo dedamosiomis;
- „Sector“ – pagal nuostatos signalo vektoriaus kampą apskaičiuoja funkcionavimo sektorių erdvinio vektoriaus moduliacijos sekstante;
- „Deg 2 d1_d2_d0“ – pagal nuostatos signalo vektoriaus ilgį apskaičiuojama reikiamos komutacijos dedamosios;
- „Symmetric sequence“ – šioje posistemėje pagal apskaičiuotas elektroninių jungiklių kombinacijos išlaikymo trukmę ir nešantį impulsų pločio moduliacijos dažnį formuojama komanda jungiklių kombinacijos pakeitimui;

- „Constant“ – blokai panaudoti išankstiniams loginėms būsenoms pagal vektorių lentelę suformuoti;
- „Multi-port switch“ – atsižvelgiant į funkcionavimo sektorių ir vektorių lentelę išskiriama jungiklių kombinacija ir nulinės, sinchroninės ir aktyvios grupių;
- „Logical operator“ - loginiai elementai elektroninių jungiklių valdymo signalams formuoti;
- „Timer“ – panaudotas modifikuotos erdvinio vektoriaus moduliacijos sektoriaus dalies įjungimui sinchronizuoti su trikdžiu energetinės sistemos įtampas imituojančiame modelyje.

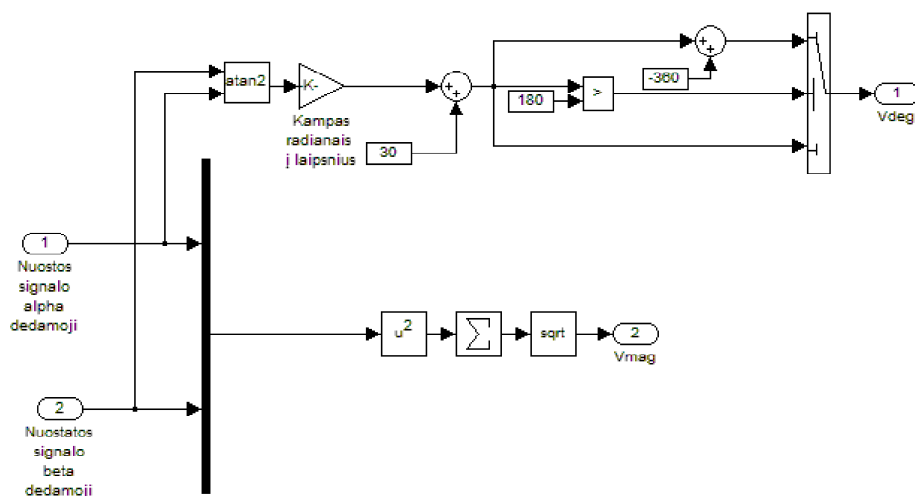


Pav. 23. „Clarko“ transformacijos posistemė

Posistemėje „Clarko transformacija“ (pav. 23) panaudoti tokie Matlab Simulink blokai:

„Demux“ – panaudotas vektorinėje formoje išreikšto nuostatos signalo išskaidymui į atskirus signalus.

„Gain“ ir „Sum“ – panaudoti matematinės operacijos su prieš tai išskaidytais nuostatos signalais.



Pav. 24. Nuostatos signalo vektoriaus kampo ir amplitudės skaičiavimo posistemė

Nuostatos signalo vektoriaus kampo ir amplitudės skaičiavimo posistemėje (pav. 24) panaudoti tokie Matlab Simulink blokai:

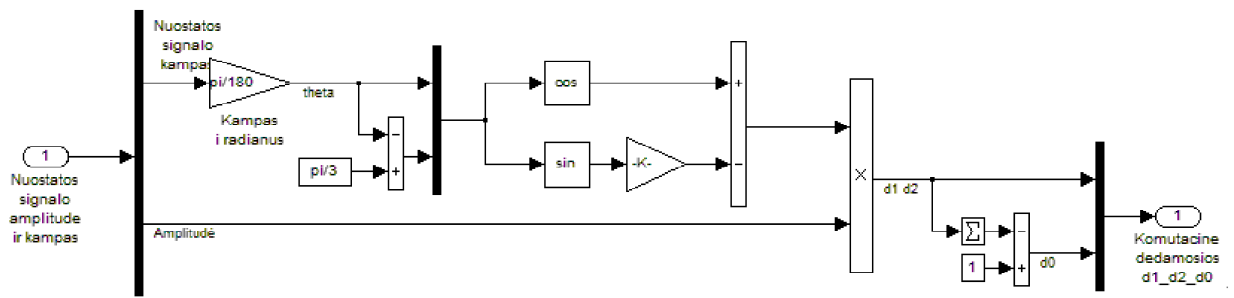
„Trigonometric function“ – panaudotas kampui iš nuostatos signalo alfa dedamosios ir nuostatos signalo beta dedamosios išskaičiavimui.

„Gain“ – panaudotas kampo išraiškos radianais vertimui į kampą laipsniais.

„Sum“ ir „Relational operation“ – nuostatos signalo vektoriaus kampui apskaičiuoti.

„Mux“ – panaudotas kelių vienodos formos signalų įėjime apjungimui į vieną vektorinį išėjimą.

„Math“ – panaudotas kartu su blokais „Sum“ ir „Gain“ nuostatos signalo vektoriaus amplitudei apskaičiuoti.

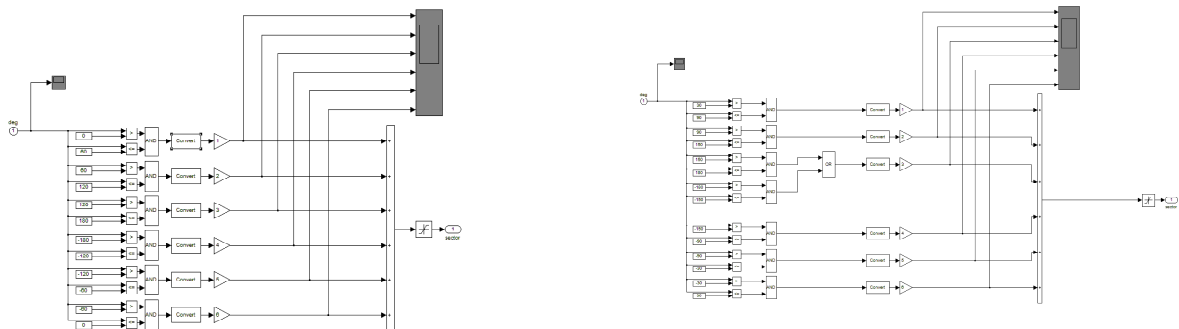


Pav. 25. Komutacijos dedamųjų skaičiavimo posistemė

Komutacijos dedamųjų (pav. 25) skaičiavimo posistemėje panaudoti tokie Matlab Simulink blokai:

„Mux“ ir „Demux“ – panaudotas signalų formato keitimui.

„Gain“, „Sum“, „Constant“, „Trigonometric functions“, „Product“ – panaudoti matematiniais veiksmais.



Pav. 26. Sektoriaus kampo formavimo posistemės: normalaus energijos tiekimo režime ir trikdžio atveju.

Sektoriaus kampo formavimo posistemėje (pav. 26) panaudoti tokie Matlab Simulink blokai: „Constant“ – formuojantis pastovų užduotą dydį. Panaudotas sektoriaus pradžios ir pabaigos kampams užduoti.

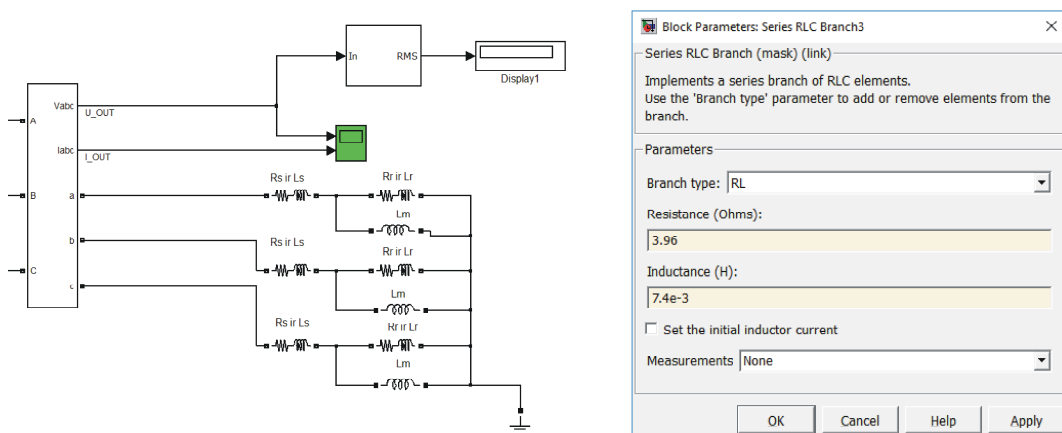
„Relational operator,“ – panaudotas sąlygos ar nuostatos signalo vektoriaus kampas patenka į blokuose „Constant“ nurodytą kampų intervalą.

„Logical operator“ – panaudoti sektoriaus numerio pagal nuostatos signalo įtampos vektoriaus kampą ir bloke „Constant“ nurodytus sektoriaus kampus formavimui.

„Data type conversion“, „Gain“, „Sum“ – panaudoti sektoriaus numerio signalo iš prieš tai suformuotos loginės išraiškos skaičiavimui.

„Saturation“ – panaudotas sektorių kiekio ribojimui.

3.4. Matricinio keitiklio apkrovos modelis



Pav. 27. Matricinio keitiklio apkrovos modelis

Matricinio keitiklio apkrovai asinchroniniu varikliu imituoti panaudota asinchroninio variklio atstojamoji schema (pav. 27) sudaryta iš Matlab Simulink blokų „Series RCL Branch“. Induktyviųjų ir aktyviųjų dedamųjų dydžiai parinkti tokie, kad imituotų 1 lentelėje pateikto asinchroninio variklio parametrus nusistovėjusiame režime.

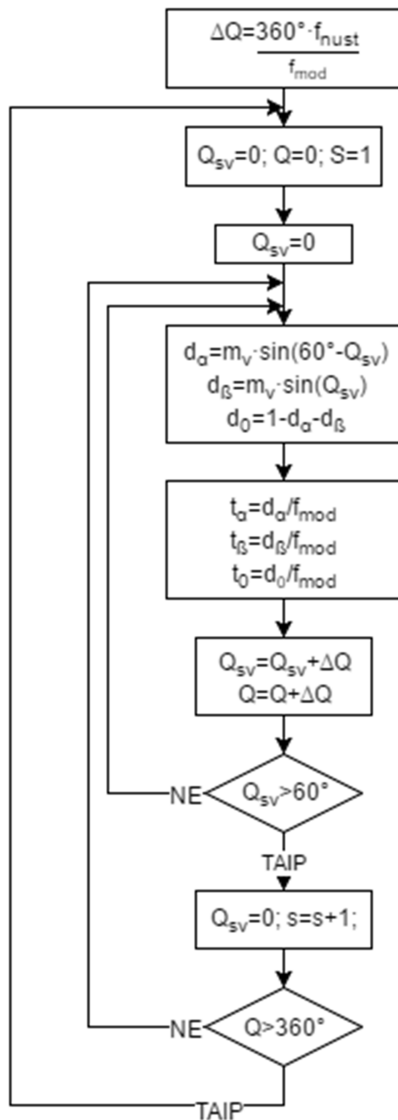
Lentelė 2. Imituoto asinchroninio variklio atstojamosios schemos parametrai.

P	0,75 kW	R_S	3,96 Ω
		R_R	2,45 Ω
U_N	230 V	L_S	7,4 mH
p	4 poliai	L_R	11,1 mH
		L_m	162mH

4. Fizinis matricinio dažnio keitiklio modelis

4.1. Erdvės vektorių moduliacijos algoritmas

Erdvės vektorių moduliacijos algoritmas pavaizduotas 28 paveiksle veikia taip: ciklo pradžioje, pagal nustatytą moduliacijos dažnį (f_{mod}) ir pasirinktą įtampos dažnį (f_{nust}) apskaičiuojamas įtampos vektoriaus žingsnis laipsniais vektorių erdvėje per kiekvieną įtampos moduliacijos periodą (ΔQ). Turint įtampos vektoriaus žingsnį, įvertinant išėjimo įtampos moduliacijos koeficientą (m_v) galima apskaičiuoti impulsų ilgį, per kuriuos išėjimo fazės bus jungiamos su atitinkamomis įėjimo fazėmis (t_α ; t_β ; t_0). Įtampos vektoriui pasiekus 60° poziciją sektoriuje (S), vektoriaus kampas (Q_{sv}) nustatomas į 0° , taip pat keičiamas sektorius, t.y. aktyvių jungimo įtampų konfigūracijos. Įtampos vektoriui pasiekus 360° kampą (Q) vektorių erdvėje, kampas nustatomas į 0° , sektorius (S) į 1. Taip ciklas kartojamas generuojant norimo dažnio įtampą.



Pav. 28 Erdvės vektorių moduliacijos algoritmas.

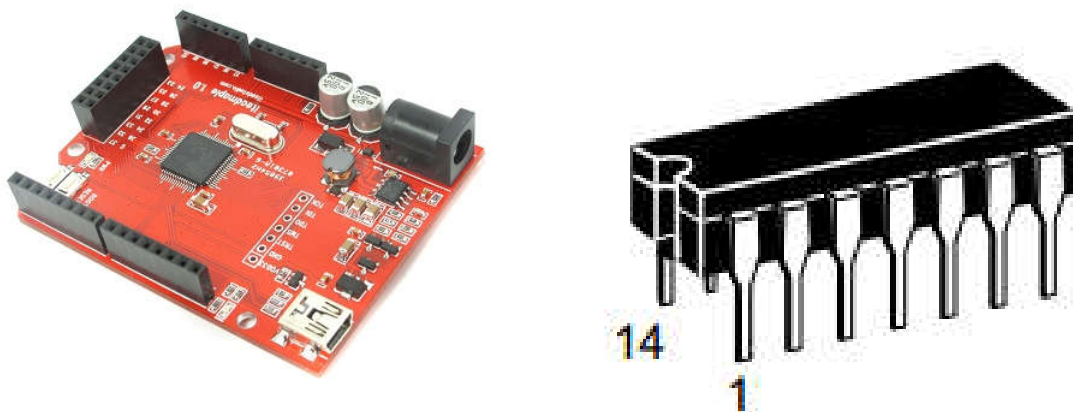
4.2. Erdvės vektorių moduliacijos modeliui naudojama įranga

Erdvės vektorių moduliacijos realizavimui parinkta „Iteadmaple v1.0“ (29 paveikslas) platforma su 72MHz ARM Cortex M3 mikrovaldikliu [18]. Ši platforma pasirinkta dėl savo spartaus ARM mikrovaldiklio, patogaus priėjimo prie fizinių išėjimų ir maitinimo galimybių bei

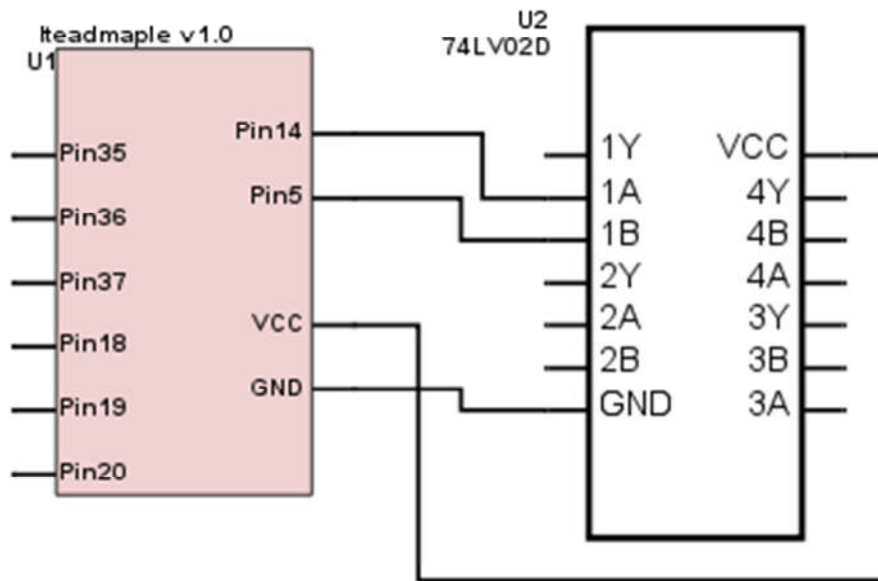
galimybės programuoti mikrovaldiklį „Arduino IDE“ pagalba, kas leidžia supaprastinti kai kurias programos dalis.

Kadangi, bet kuriuo metu yra aktyvus bent vienas mikrovaldiklio išėjimas įtampos vektoriaus formavimui, impulsų generavimui taip pat panaudotas loginis „arba-ne“ elementas „74LS02“ (29 paveikslas), tai leidžia sumažinti mikrovaldiklio apkrovimą.

Iš viso panaudoti 8 diskretiniai išėjimai, 2 – įtampos vektoriaus moduliavimui, 6 – sektoriaus jungimui.



Pav. 29 “Iteadmaple v1.0” platforma – kairėje, “74LS02” loginis elementas – dešinėje



Pav. 30 „Iteadmaple v1.0“ ir „74LS02“ jungimo schema

Sektorių 1-6 išėjimai atitinkamai – pin35, 36, 37, 18, 19, 20, t_a impulsas išvedamas per pin14, t_b – pin5, t_0 – per loginio elemento „74LS02“ 1Y išėjimą.

4.3. Erdvės vektorių moduliacijos programa

„Iteadmaple v1.0“ programavimui naudojamas „Arduino IDE“ paketas.

Programos pradžioje aprašomi programoje naudojami kintamieji ir funkcijos:

```
void handler(void); //moduliacijos periodo pradžios pertraukimas
void handler3(void); //ta signalo pabaigos ir tb - pradžios pertraukimas
void handler4(void); //tb signalo pabaigos pertraukimas

float frmod = 9900; //moduliacijos dažnis
float periodas; //moduliacijos periodas mikrosekundėmis
float frnust = 35; //nustatytas dažnis
float dq; //kampo pokytis per moduliacijos periodą
float qsv = 0; //kampas sektoriuje
float mv = 1; //išėjimo įtampos moduliacijos koeficientas
byte s = 1; //sektorius numeris
int tans[64]; //impulsų laikai mikrosekundėmis
int tbns[64];
int ta[64]; //impulsu laikai apskaičiuoti įvertinant "prescaler"
int tb[64];

byte i = 0; //impulso eilės numeris sektoriuje
byte imax; //impulsų skaičius sektoriuje
int sine60[64]; //apskaičiuotos reikalingos sinusų ir kosinusų reikšmės
int cos60[64];
```

```
bool sector;//sektorius perjungimo indikatorius
bool pinas;//impulsų trukmės užkrovimo indikatorius
```

Funkcijoje „setup“ aprašomi reikalingi išėjimai. Tam naudojama „Arduino“ funkcija pinMode(pin, mode), kur pin – norimas apsašyti išėjimas ar įėjimas aprašant išėjimų portą ir išėjimo eilės numerį porte, mode – „INPUT“ – įėjimas, „OUTPUT“ – išėjimas:

```
pinMode(PB8, OUTPUT);//pin14
pinMode(PB6, OUTPUT);//pin5

pinMode(PC6, OUTPUT);//pin35
pinMode(PC7, OUTPUT);//pin36
pinMode(PC8, OUTPUT);//pin37
pinMode(PC3, OUTPUT);//pin18
pinMode(PC4, OUTPUT);//pin19
pinMode(PC5, OUTPUT);//pin20
```

Toliau apskaičiuojamas moduliacijos periodas mikrosekundėmis ir vektoriaus žingsnis sektoriuje:

```
periodas = 1000000 / frmod;//us
dq = 360 * frnust / frmod;
```

Apskaičiuojamos sinusų ir kosinusų vertės pagal moduliacijos žingsnius, programoje patogiau naudoti moduliacijos žingsnius nei kampą, taip mažiau apkraunamas mikrovaldiklis, o programos moduliacijos tikslumui įtakos nėra. Sinusų ir kosinusų verčių skaičiavimui naudojama „Arduino“ funkcija „sin“, norint sutaupyti mikrovaldiklio programinę atmintį, vertės išsaugomos „integer“ kintamojo tipu, išlaikant tikslumą padauginamos iš 1000:

```
while (qsv <= 60) {
    sine60[i] = int(1000 * sin(dq * i / 57.2958));
    cos60[i] = int(1000 * sin((60 - dq * i) / 57.2958));
    i++;
    qsv = qsv + dq;
}
```

Naudojant sektoriaus moduliacijos žingsnių eilės numerius bei apskaičiuotas sinusų ir kosinusų reikšmes apskaičiuojamos impulsų trukmės. Norint išlaikyti tikslumą, vertės skaičiuojamos šimtosiomis mikrosekundės dalimis:

```
qsv = 0;
imax = i - 1;
for (i = 0; i <= imax; i = i + 1) {
    tans[i] = round(periodas / 10 * mv * cos60[i]);
    tbns[i] = round(periodas / 10 * mv * sine60[i]);
}
```

Toliau apskaičiuojami laikai pertraukimų aktyvavimui įvertinant įvestą „prescaler“:

```
for (i = 0; i <= imax; i = i + 1) {
```

```

ta[i] = tans[i] * 2 / 100 - 1;
if (ta[i] == 0) {
    ta[i] = 6;
}
tb[i] = tbns[i] * 2 / 100 + ta[i] - 1; //ns/10
if (tbns[i] == 0) {
    tb[i] = tb[i] + 7;
}
}
}

```

Sukonfigūrojami reikalingi pertraukimai.

Timer4.setChannel1Mode(TIMER_OUTPUTCOMPARE); - nustatomas pertraukimo tipas, šiuo atveju pertraukimo laikas skaičiuojamas lyginant su iš anksto nustatyta reikšme ir šiai sutapus aktyvuojamas pertraukimas.

Timer4.pause(); - laikinai išjungiamas pertraukimas, tam kad pertraukimai nebūtų generuojami jų dar nesukonfigūravus.

Timer4.setCompare(TIMER_CH1, 1); - nustatomas pertraukimo naudojamas kanalas.

Timer4.setOverflow(periodas * 2 - 1); - nustatomas laikas pertraukimų generavimui, "Timer4" naudojamas moduliacijos periodo pradžios veiksmams atlikti, dėl to iš karto nustatomas moduliacijos periodo trukmei, įvertinant "prescaler" vertę.

Timer4.setPrescaleFactor(36);/"Prescaler" nustatymas, "prescaler" nurodo, koku dažniu skaičiuojamas pertraukimo laikas, "prescaler" išreiškia, kas kiek mikrovaldiklio ciklų pertraukimo laikmatis padidės vienetu, šiuo atveju „prescaler“ nustatymas 36, reiškia, kad laikmačio vertė padidės vienetu kas 0,5 mikrosekundės, taigi „prescaler“ nurodo laikmačio tikslumą.

Timer4.attachCompare1Interrupt(handler4); - ši funkcija priskiria pertraukimą reikalingai funkcijai.

```

Timer4.setChannel1Mode(TIMER_OUTPUTCOMPARE);
Timer4.pause();
Timer4.setCompare(TIMER_CH1, 1);
Timer4.setOverflow(periodas * 2 - 1);
Timer4.setPrescaleFactor(36);
Timer4.attachCompare1Interrupt(handler4);

Timer2.setChannel1Mode(TIMER_OUTPUTCOMPARE);
Timer2.pause();
Timer2.setCompare(TIMER_CH1, 1);
Timer2.setPrescaleFactor(36);
Timer2.attachCompare1Interrupt(handler);

```

```

Timer3.setChannel1Mode (TIMER_OUTPUTCOMPARE);
Timer3.pause ();
Timer3.setCompare (TIMER_CH1, 1);
Timer3.setPrescaleFactor (36);
Timer3.attachCompare1Interrupt (handler3);

```

Programoje naudojami trys pertraukimai, „Timer4“ naudojamas moduliacijoms periodams generuoti, o kiti du – „Timer2“ ir „Timer3“ impulsų generavimui, kad pastarieji du pertraukimai nebūtų generuojami daug kartų moduliacijos periodo metu, jų „CR1“ registų bitai „OPM“ nustatomi į vienetą [19].

```

TIMER2_BASE->CR1 = 0x0019;
TIMER3_BASE->CR1 = 0x0019;

```

Nenorint, kad atsirastų nepageidaujami sisteminiai pertraukimai, kurie galėtų neleisti suveikti moduliacijai reikalingiems pertraukimams, išjungiamo mikrovaldiklio sąsaja per USB ir integruotas, sisteminis laiko skaičiavimas:

```

Serial.end ();
systick_disable ();

```

Prieš paleidžiant moduliaciją aktyvuojami pertraukimai:

```

Timer2.resume ();
Timer3.resume ();
Timer4.resume ();

```

Pagrindinėje programos funkcijoje „loop()“ vykdomas pertraukimų užkrovimas pagal reikalingą žingsnį sektoriuje, tam pertraukimo laikas įrašomas tiesiai į mikrovaldiklio registrą (TIMER2_BASE->ARR = ta[i];) [19] nenaudojant „Iteastudio“ sukurtų funkcijų, taip įrašymas vyksta kur kas greičiau. Tam, kad pertraukimai būtų užkrauti tik vieną kartą moduliacijos periodo metu, naudojamas indikatorius „pinas“, jei t_0 impulsas trumpesnis nei viena mikrosekundė jis ignoruojamas, nes tokie trumpi impulsai neturi įtakos įtampos generavimui, veltui apkraunamas mikrovaldiklis:

```

if (pinas == 1) {
    TIMER2_BASE->ARR = ta[i]; //overflow nustatymas
    TIMER3_BASE->ARR = tb[i];
    i++;
    pinas = 0;
    if (tb[i] > 198) {

```

```

    TIMER3_BASE->CR1 = 0x0018;
}
}

```

Sektorių perjungimas vykdomas po paskutinio žingsnio sektoriuje, tam naudojamas „switch“ funkcija. Tam, kad sektoriaus perjungimas būtų kuo tikslesnis, vietoje „Iteastudio“ sukurtų funkcijų, išėjimai valdomi į mikrovaldiklio registrus tiesiai įrašant bitus ((GPIOC_BASE)->BRR = BIT(8));, BRR – išėjimo nustatymas į loginį nulį, BSRR - vieneta:

```

if (i == imax) {
    i = 0;
    sector = 1;
}
if (i == 2 && sector == 1) {
    ++s;
    if (s == 7)
    {
        s = 1;
    }
    switch (s) {
        case 1:
            (GPIOC_BASE)->BRR = BIT(8);
            (GPIOC_BASE)->BSRR = BIT(3);
            break;
        case 2:
            (GPIOC_BASE)->BRR = BIT(3);
            (GPIOC_BASE)->BSRR = BIT(4);
            break;
        case 3:
            (GPIOC_BASE)->BRR = BIT(4);
            (GPIOC_BASE)->BSRR = BIT(5);
            break;
        case 4:
            (GPIOC_BASE)->BRR = BIT(5);
            (GPIOC_BASE)->BSRR = BIT(6);
            break;
        case 5:
            (GPIOC_BASE)->BRR = BIT(6);
            (GPIOC_BASE)->BSRR = BIT(7);
            break;
        case 6:
            (GPIOC_BASE)->BRR = BIT(7);
            (GPIOC_BASE)->BSRR = BIT(8);
            break;
    }
    sector = 0;
}
}

```

Impulsų moduliacijai naudojami aprašyti pertraukimai:

```

void handler4(void) {
    (GPIOB_BASE)->BRR = BIT(6);
}

```



```

TIMER3_BASE->CR1 = 0x0019;
TIMER2_BASE->CR1 = 0x0019;
(GPIOB_BASE)->BSRR = BIT(8);
pinas = 1;
}

void handler(void) {
    (GPIOB_BASE)->BRR = BIT(8); //pin14
    (GPIOB_BASE)->BSRR = BIT(6); //pin5
}

void handler3(void) {
    (GPIOB_BASE)->BRR = BIT(6); //pin5
}

```

4.4. Fizinio modelio rezultatai

Realizavus algoritmą valdiklyje ir išbandžius su matriciniu dažnio keitikliu gautos oscilogramos duomenys perkelti į kompiuterį. MATLAB Simulink bloko „THD“ pagalba išmatuotas šalutinių harmonikų kiekis išėjimo įtampoje, THD = 4,65 %. Matricinio dažnio keitiklio Simulink modelio THD = 8 %. Efektinės įtampos vertės Simulink modelyje – 100 V ir fiziniame modelyje – 94 V, esant 110 V maitinimo įtampai.

Simulink modelyje nevertinami realūs elektroninių jungiklių parametrai, todėl gaunama didesnė efektinė įtampos vertė ir didesnis šalutinių harmonikų kiekis. Sukurtas Simulink modelis adekvatus.

5. Modeliavimo rezultatai

Apjungus praeitame skyriuje aprašytus trikdžius energetinėje sistemoje imituojantį modelį, matricinio keitiklio jėgos grandinių modelį ir erdvinio vektoriaus moduliacijos (EVM) modelį ir matricinio keitiklio apkrovos modelį sudarytas bendras sistemos Matlab Simulink modelis. Atlikus tyrimą modeliavimo būdu nustatytas efektas, kurį padaro erdvinio vektoriaus moduliacijos sektoriaus modifikacija. Modeliavimo rezultatai pateikti lentelėje 3.

Lentelė 3. Matricinio keitiklio išėjimo įtampos efektinė vertė įtampos trikdžio atveju.

Trikdžio tipas	Matricinio keitiklio išėjimo įtampos efektinė vertė, V					
	A fazėje		B fazėje		C fazėje	
	Tradicinė EVM	Modifikuota EVM	Tradicinė EVM	Modifikuota EVM	Tradicinė EVM	Modifikuota EVM
1	149	230	128	192	117	192
2	149	210	140	173	117	167
3	205	193	175	155	154	148
4	197	159	197	137	136	98

Lentelė 4. Matricinio keitiklio išėjimo įtampos šalutinių harmonikų kiekis įtampos trikdžio atveju

Trikdžio tipas	Matricinio keitiklio išėjimo įtampos šalutinių harmonikų kiekis, %					
	A fazėje		B fazėje		C fazėje	
	Tradicinė EVM	Modifikuota EVM	Tradicinė EVM	Modifikuota EVM	Tradicinė EVM	Modifikuota EVM
1	127	54,9	61,4	36,1	58,7	54,8
2	59,5	71,2	74,8	61,7	208	91,9
3	49,1	79,5	42,2	51,2	78,5	65,04
4	51,05	72,08	156,6	139,4	51,1	118

Lentelė 5. Erdvinio vektoriaus moduliacijos sektoriaus modifikacijos įtaka išėjimo įtampos efektinei vertei.

Trikdžio tipas	Matricinio keitiklio išėjimo įtampos efektinės vertės pokytis, %		
	A fazėje	B fazėje	C fazėje
1	54	50	64
2	41	24	43
3	-6	-11	-4
4	-19	-30	-28

Lentelė 6. Erdvinio vektoriaus moduliacijos sektoriaus modifikacijos įtaka išėjimo įtampos šalutinių harmonikų kiekiui.

Trikdžio tipas	Matricinio keitiklio išėjimo įtampos šalutinių harmonikų kiekio pokytis, %		
	A fazėje	B fazėje	C fazėje
1	-57	-41	-7
2	20	-18	-56
3	62	21	-17
4	41	-11	131

Išvados

1. Dėl reaktyviųjų energijos kaupimo elementų nebuvimo tiesioginiams matriciniams keitikliams netinka įtampos trikdžių kompensavimo būdai, kurie taikomi tradiciniams dažnio keitikliams su tarpine nuolatinės srovės grandimi. Literatūros šaltiniuose siūlomi trikdžių matriciniuose keitikliuose kompensavimo būdai susiję su moduliacijos indekso arba nuostatos signalo padidinimu netinka Lietuvos energetinėje sistemoje pasitaikantiems trikdžiams, kuriu metu įvyksta įtampų nesimetriškumas. Šiems trikdžiams kompensuoti gali būti taikomas erdvinio vektoriaus formavimo segmentų adaptacija paremtas metodas.
2. Atlikus tyrimą Matlab Simulink modeliu nustatyta, kad susiejant erdvinės vektorių moduliacijos segmentų kampą su elektros energijos tiekimo linijų įtampų fazės kampu trikdžio atveju galima padidinti išėjimo įtampos efektinę vertę nuo 50 % iki 64 % 1 tipo trikdžio atveju, nuo 24 % iki 43 % 2 tipo trikdžio atveju, lyginant su tradicine erdvinio vektoriaus moduliacija. Tačiau 3 ir 4 tipo trikdžių atveju toks kampo perstūmimas lemia įtampų efektinės vertės sumažėjimą.
3. Panaudojus Matlab Simulink modeliavimą nustatyta, kad susiejant erdvinės vektorių moduliacijos segmentų kampą su elektros energijos tiekimo linijų įtampų fazės kampu trikdžio atveju galima sumažinti išėjimo įtampų šalutinių harmonikų kiekį kai kuriose fazėse nuo 7 % iki 57 % 1 tipo trikdžio atveju, nuo 18 % iki 56 % 2 tipo trikdžio atveju, lyginant su tradicine erdvinio vektoriaus moduliacija. Tačiau 3 ir 4 tipo trikdžių atveju toks kampo perstūmimas lemia šalutinių harmonikų kiekio daugumoje fazių padidėjimą.

Literatūros sąrašas

- [1] S. Gečys, A. Kalvaitis, P. Smolskas, Transformatoriai, asinchroninės mašinos, Kaunas: "Technologija", 2012.
- [2] P. Szczesniak, Three-Phase AC–AC Power Converters Based on Matrix Converter Topology, Londonas: Springer-Verlag, 2013.
- [3] Kai Sun, Daning Zhou, Lipei Huang, Kouki Matsuse, "Compensation control of matrix converter fed induction motor drive," in 39th IAS Annual Meeting, 2004.
- [4] Jun-Koo Kang, Hidenori Hara, Ahmet M. Hava, Eiji Yamamoto, Eiji Watanabe, Tsuneo Kume, "The Matrix Converter Drive Performance Under Abnormal Input Voltage Conditions," IEEE TRANSACTIONS ON POWER ELECTRONICS, pp. 721-730, Rugsėjis 2002.
- [5] Jun Oyama, Xiarong Xia, Tsuyoshi Higuchi, Eiji Yamada, "Displacement Angle Control of Matrix Converter," IEEE, pp. 1033-1039, 1997.
- [6] D. Vincenti, P.D. Ziogas, and R.V. Patel, "An analysis and design of a force commutated three-phase PWM AC controller with input unbalance correction capability," IEEE, pp. 487-493, 1992.
- [7] P. W. Wheeler, J. Rodriguez, J. C. Clare, and L. Empringham, "Matrix converter: A technology review," IEEE Transactions on Industrial Electronics, Vol. 49, No. 2, p. 276–288, Balandis 2002.
- [8] Prasad, Rashmi. "Low Voltage Ride-Through Capability for Matrix Converter fed Adjustable-Speed Induction Machine Drives for Industrial and Wind Applications" A dissertation submitted to the faculty of the graduate school of the University of Minnesota, 2011.
- [9] Han Ju Cha, Prasad N. Enjeti, "A New Ride-through Approach for Matrix Converter Fed Adjustable Speed Drives," IEEE, pp. 2555-2561, 2002.
- [10] Sarigshin Kwak, Rami M.O. Dabbousi, Hamid A. Toliyat, "An Instantaneous Compensation Technique for Supply Disturbances in Direct Frequency Converters," IEEE, pp. 956-961, 2002.
- [11] Iglesias RL, Arantegui RL, Alonso MA. Power electronics evolution in wind turbines - A market-based analysis. Renew Sust Energ Rev 2011; 15: 4982–4993.
- [12] Vasipalli V, Phulambrikar SP, Agrawal A. Power Quality Improvement in DFIG System with Matrix Converter in Wind Energy Generation with Space Vector Control Techniques. In: 2015

International Conference on Technological Advancements in Power and Energy (TAP Energy); 24-26 June 2015; Kollam, India: IEEE. pp. 73–78.

[13] Andreu J, Kortabarria I, Ormaetxea E, Ibarra E, Martin JL, Apiñaniz S. A step forward towards the development of reliable matrix converters. *IEEE T Ind Electron* 2012; 59 (1): 167–183.

[14] Friedli T; Kolar JW; Rodriguez J; Wheeler PW. Comparative Evaluation of Three-Phase AC–AC Matrix Converter and Voltage DC-Link Back-to-Back Converter Systems. *IEEE T Ind Electron* 2012; 59 (12): 4487–4510.

[15] Svinkunas G, Deksnys R. Influence of voltage sags for frequency converters. In: *International Conference on Electrical and Control Technologies*; 8-9 May 2008; Kaunas, Lithuania: pp. 243 – 246.

[16] Shahmohamadi I, Feyzi MR, Babaei E. Simultaneous implementation of minimum torque pulsation and unity power factor in matrix converter drive. In: *8th Iranian Conference on Electrical Engineering*; 11-13 May 2010; Isfahan, Iran.

[17] Sun K, Zhou D, Huang L, Matsuse K. Compensation control of matrix converter fed induction motor drive under abnormal input voltage conditions. In: *Industry Applications Conference*, 3-7 October 2004; Seattle, Washington, USA: IEEE. pp. 623-630.

[18] Iteadstudio, "Iteadmaple v1.0 datasheet," 2011.

[19] STMicroelectronics, "RM0008 Reference manual," 2011.

Priedai

1. Programos kodas:

```
2. void handler(void); //moduliacijos periodo pradžios pertraukimas
3. void handler3(void); //ta signalo pabaigos ir tb - pradžios pertraukimas
4. void handler4(void); //tb signalo pabaigos pertraukimas
5.
6. float frmod = 9900; //moduliacijos dažnis
7. float periodas; //moduliacijos periodas us
8. float frnust = 35; //nustatytas dažnis
9. float dq; //kampo pokytis per moduliacijos perioda
10. float qsv = 0; //kampas sektoriuje
11. float mv = 1; //isejimo itamos moduliacijos koef.
12. byte s = 1; //sektoriaus numeris
13. int tans[64]; //impulsu laikai mikrosekundemis
14. int tbns[64];
15. int ta[64]; //impulsu laikai apskaiciuoti ivertinant "prescaler"
16. int tb[64];
17.
18. byte i = 0; //impulso eiles numeris sektoriuje
19. byte imax; //impulsu skaicius sektoriuje
20. int sine60[64]; //apskaiciuotos reikalingos sinusu ir kosinusu reiksmes
21. int cos60[64];
22.
23. bool sector; //sektoriaus perjungimo indikatorius
24. bool pinas; //impulsu trukmes uzskrovimo indikatorius
25.
26. void setup() {
27.     // Set up the LED to blink
28.     pinMode(PB8, OUTPUT); //pin14
29.     pinMode(PB6, OUTPUT); //pin5
30.
31.     pinMode(PC6, OUTPUT); //pin35
32.     pinMode(PC7, OUTPUT); //pin36
33.     pinMode(PC8, OUTPUT); //pin37
34.     pinMode(PC3, OUTPUT); //pin18
35.     pinMode(PC4, OUTPUT); //pin19
36.     pinMode(PC5, OUTPUT); //pin20
37.
38.     periodas = 1000000 / frmod; //us
39.     dq = 360 * frnust / frmod;
40.
41.     while (qsv <= 60) {
42.         sine60[i] = int(1000 * sin(dq * i / 57.2958));
43.         cos60[i] = int(1000 * sin((60 - dq * i) / 57.2958));
44.         i++;
45.         qsv = qsv + dq;
46.     }
47.
48.     qsv = 0;
49.     imax = i - 1;
50.     for (i = 0; i <= imax; i = i + 1) {
51.         tans[i] = round(periodas / 10 * mv * cos60[i]); //ns/10
52.         tbns[i] = round(periodas / 10 * mv * sine60[i]); //ns/10
53.     }
54.
55.
```

```

56.
57.   for (i = 0; i <= imax; i = i + 1) {
58.       ta[i] = tans[i] * 2 / 100 - 1;
59.       if (ta[i] == 0) {
60.           ta[i] = 6;
61.       }
62.       tb[i] = tbns[i] * 2 / 100 + ta[i] - 1; //ns/10
63.       if (tbns[i] == 0) {
64.           tb[i] = tb[i] + 7;
65.       }
66.   }
67.   i = 0;
68.
69.   pinas = 1;
70.
71.   // Set up an interrupt on channel 1
72.   Timer2.setChannell1Mode(TIMER_OUTPUTCOMPARE);
73.   Timer2.pause();
74.   Timer2.setCompare(TIMER_CH1, 1);
75.   Timer2.setOverflow(5);
76.   Timer2.setPrescaleFactor(36);
77.   Timer2.attachCompare1Interrupt(handler);
78.
79.   Timer3.setChannell1Mode(TIMER_OUTPUTCOMPARE);
80.   Timer3.pause();
81.   Timer3.setCompare(TIMER_CH1, 1);
82.   Timer3.setOverflow(190);
83.   Timer3.setPrescaleFactor(36);
84.   Timer3.attachCompare1Interrupt(handler3);
85.
86.   Timer4.setChannell1Mode(TIMER_OUTPUTCOMPARE);
87.   Timer4.pause();
88.   Timer4.setCompare(TIMER_CH1, 1);
89.   Timer4.setOverflow(periodas * 2 - 1);
90.   Timer4.setPrescaleFactor(36);
91.   Timer4.attachCompare1Interrupt(handler4);
92.
93.   TIMER2_BASE->CR1 = 0x0019;
94.   TIMER3_BASE->CR1 = 0x0019;
95.
96.   Serial.end();
97.   systick_disable();
98.   // Start the timer counting
99.   Timer2.resume();
100.  Timer3.resume();
101.  Timer4.resume();
102. }
103.
104. void loop() {
105.     if (pinas == 1) {
106.         TIMER2_BASE->ARR = ta[i]; //overflow nustatymas
107.         TIMER3_BASE->ARR = tb[i];
108.         i++;
109.         pinas = 0;
110.         if (tb[i] > 198) {
111.             TIMER3_BASE->CR1 = 0x0018;
112.         }
113.     }
114. }

```



```

115.  if (i == imax) {
116.      i = 0;
117.      sector = 1;
118.  }
119.  if (i == 2 && sector == 1) {
120.      ++s;
121.      if (s == 7)
122.      {
123.          s = 1;
124.      }
125.      switch (s) {
126.          case 1:
127.              (GPIOC_BASE)->BRR = BIT(8);
128.              (GPIOC_BASE)->BSRR = BIT(3);
129.              break;
130.          case 2:
131.              (GPIOC_BASE)->BRR = BIT(3);
132.              (GPIOC_BASE)->BSRR = BIT(4);
133.              break;
134.          case 3:
135.              (GPIOC_BASE)->BRR = BIT(4);
136.              (GPIOC_BASE)->BSRR = BIT(5);
137.              break;
138.          case 4:
139.              (GPIOC_BASE)->BRR = BIT(5);
140.              (GPIOC_BASE)->BSRR = BIT(6);
141.              break;
142.          case 5:
143.              (GPIOC_BASE)->BRR = BIT(6);
144.              (GPIOC_BASE)->BSRR = BIT(7);
145.              break;
146.          case 6:
147.              (GPIOC_BASE)->BRR = BIT(7);
148.              (GPIOC_BASE)->BSRR = BIT(8);
149.              break;
150.      }
151.      sector = 0;
152.  }
153. }
154.
155. void handler4(void) {
156.     (GPIOB_BASE)->BRR = BIT(6);
157.     TIMER3_BASE->CR1 = 0x0019;
158.     TIMER2_BASE->CR1 = 0x0019;
159.     (GPIOB_BASE)->BSRR = BIT(8);
160.     pinas = 1;
161. }
162.
163. void handler(void) {
164.     (GPIOB_BASE)->BRR = BIT(8); //pin14
165.     (GPIOB_BASE)->BSRR = BIT(6); //pin5
166. }
167.
168. void handler3(void) {
169.     (GPIOB_BASE)->BRR = BIT(6); //pin5
170. }

```