

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

**Arnoldas Pranka**

**GEDIMO VIETOS NUSTATYMO ALGORITMŲ TYRIMAS**

Baigiamasis magistro projektas

**Vadovas**

Doc. dr. Almantas Bandza

**KAUNAS, 2018**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

**ELEKTROS ENERGETIKOS SISTEMŲ KATEDRA**

**GEDIMO VIETOS NUSTATYMO ALGORITMŲ TYRIMAS**

Baigiamasis magistro projektas

**Elektros energetikos sistemos (kodas 621H63005)**

**Vadovas**

Doc. dr. Almantas Bandza

**Recenzentas**

Prof. dr. Alfonsas Morkvėnas

**Projektą atliko**

Arnoldas Pranka

**KAUNAS, 2018**



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Elektros ir elektronikos fakultetas

---

(Fakultetas)

Arnoldas Pranka

---

(Studento vardas, pavardė)

Elektros energetikos sistemos, 621H63005

---

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Gedimo vietos nustatymo algoritmų tyrimas“

**AKADEMINIO SAŽININGUMO DEKLARACIJA**

20 \_\_\_\_\_ 18 \_\_\_\_\_ m. \_\_\_\_\_ 05 \_\_\_\_\_ 24 \_\_\_\_\_ d.

\_\_\_\_\_ Kaunas

Patvirtinu, kad mano **Arnoldo Prankos** baigiamasis projektas tema „Gedimo vietos nustatymo algoritmų 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)

Pranka, Arnoldas. Gedimo vietos nustatymo algoritmų tyrimas. *Magistro* baigiamasis projektas / vadovas doc. dr. Almantas Bandza; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Elektros energetikos sistemų katedra.

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

Reikšminiai žodžiai: *Gedimo vieta, trumpieji jungimai, viršįtampiai.*

Kaunas, 2018. 69 p.

## SANTRAUKA

Elektros energetikos sistemos patikimumas bei nepertraukiamas elektros energijos tiekimas elektros vartotojams yra vienas svarbiausių veiksnių elektros energetikos sistemos darbe. Kaip žinoma perdavimo ir skirstymo tinkluose elektros energijos tiekimas dažnai nutrūksta dėl trumpalaikių bei ilgalaikių gedimų. Šiuos gedimus sukelia viršįtampiai, laikini trumpieji jungimai, sistemos perkrovimai ar mechaniniai pažeidimai elektros sistemose, jų linijose ar kitose jos dalyse. Gedimai gali pasišalinti savaime, bet jei avarija sudėtingesnė, gedimui sutvarkyti reikia vykdyti fiziškai į vietą ir jį pašalinti. Gedimo vietos nustatymas yra vienas svarbiausių veiksnių greitam ir efektyviam gedimo pašalinimui. Todėl gedimo vietai nustatyti elektros energetikos sistemoje yra pasitelkiami įvairūs metodai. Vienas iš jų yra gedimo vietos nustatymo algoritmų panaudojimas. Pasitelkiant algoritmus nustatyti tiksliai gedimo vietai naudojamos įvairios programos. Magistrinio darbo pagrindinis tikslas yra sukurti bei ištyri platformą, kuri galėtų gedimo vietos nustatymo algoritmų pagalba surasti kuo tikslesnę gedimo vietą. Bei ištyri ar sukurta platforma gali būti panaudota elektros energetikos sistemoje.

Taigi šiame tyrime sukūriau platformą, kuri padėtų nustatyti gedimo vietą algoritmų pagalba. Algoritmų sudarymui pasitelkiau „MATLAB“ kompiuterinę programą. Šiame darbe sukūriau modelį, kuris sugeba gana tiksliai nustatyti gedimo vietą. Šį modelį bei keletą pritaikytų algoritmų ištyriau su realiais duomenimis. Taip pat ištyriau modelio tikslumą bei jo greitį su sumodeliuotos sistemos duomenimis. Šią elektros energetikos sistemą sumodeliavau Simulink simuliacinės programos pagalba.

Pagal gautus rezultatus ištyriau mano pateikiamos platformos galimybes bei pritaikymą elektros energetikos sistemoje.

Pranka, Arnoldas. Title of the final project *Research of Fault Location Algorithms: Master's thesis in Masters degree in Electrical Energy* / supervisor assoc. prof. Almantas Bandza. Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, Department of Electric Power Systems.

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

Key words: *Fault place, short circuits, over-voltage.*

Kaunas, 2018. 69 p.

## **SUMMARY**

Reliability of the power system and uninterrupted power supply to electricity consumers is one of the most important factors in the work of the power system. As it is known power supply in transmission and distribution networks, often breaks down due to short-term and long-term failures. These failures are caused by overvoltages, temporary short circuits, system overloads or mechanical damage in electrical systems, their lines or in other parts of it. Faults can escape by itself, but if the crash is more complicated to fix the fault it is necessary to go physically to the place and remove it. Fault location is one of the most important factors for fast and efficient fault removing. Therefore, a variety of methods are used in the power system to determine the location of a fault. One of them is to use the fault location algorithms. By using algorithms to determine precisely the failure area, various programs are used. The main purpose of the master's thesis is to create and explore a platform that can find the most precise fault location with the help of troubleshooting algorithms. And to explore if developed platform can be used in the power system.

In this research I created a platform that would help to identify the fault location by using algorithms. I used the MATLAB computer program to create algorithms. In this research I have created a model that is able to fairly accurately determine the fault location. I tested this model and some of the algorithms that I have applied with real data. I also examined the accuracy of the model and its speed with the data of the simulated system. I have simulated this power system with Simulink simulation program.

Based on the results I I have explored the capabilities and application of my platform to the power system.

# TURINYS

SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS.....	10
ĮVADAS.....	11
1. GEDIMO VIETOS NUSTATYMO TIKSLAS IR SVARBA.....	13
2. GEDIMO VIETOS NUSTATYMO ALGORITMAI.....	14
2.1. Gedimo vietos nustatymas paremtas pilnutinės varžos matavimu .....	14
2.2. Gedimo vietos nustatymas naudojant DC dedamųjų reikšmių pašalinimą.....	16
3. GEDIMO VIETOS NUSTATYMAS PAGAL LINIJOS ĮTAMPŲ IR SROVIŲ PEREINAMUOSIUS PROCESUS GEDIMO METU .....	23
3.1. „COMTRADE“ standarto pritaikymas „MATLAB“ programoje .....	23
3.2. Gedimo vietos nustatymas „MATLAB“ programos pagalba .....	25
3.3. Gedimo vietos nustatymas pašalinant nuolatinės srovės dedamąsias.....	30
4. SIMULINK MODELIAVIMAS.....	36
4.1. Gedimo vietos algoritmų tikslumo, paklaidos ir greičio skaičiavimų bandymai.....	38
4.1.1. Bandymas Nr.1 gedimas fazė – žemė be DC filtro.....	38
4.1.2. Bandymas Nr.2 gedimas fazė – žemė su DC filtru.....	41
4.1.3. Bandymas Nr.3 gedimas fazė – žemė su DC filtru kai nežinoma gedimo varža.....	44
4.1.4. Bandymas Nr.4 gedimas fazė – žemė be DC filtro kai gedimo vietos atstumas skaičiuojamas pagal linijos kampą .....	47
4.1.5. Bandymas Nr.5 gedimas fazė – žemė su DC filtru kai gedimo vietos atstumas skaičiuojamas pagal linijos kampą .....	50
4.1.6. Bandymas Nr.6 gedimas fazė – žemė su DC filtru metodika 2.....	53
4.2. Gedimo vietos algoritmo tikslumo vertinimas kintant gedimo pobūdžiui.....	56
4.2.1. Gedimas fazė-fazė-žemė bei fazė-fazė .....	56
5. GEDIMO VIETOS NUSTATYMO ALGORITMO MODELIO ANALIZĖ .....	61
5.1. Modelio tikslumas.....	61
5.2. Modelio greitis .....	62
5.3. Modelio skaičiavimų paklaida .....	63
6. ALGORITMO PRIVALUMAI, TRŪKUMAI BEI PRITAIKYMAS ELEKTROS ENERGETIKOS SISTEMOJE .....	65
6.1. Algoritmo privalumai ir trūkumai.....	65
6.2. Galimybės pritaikyti elektros energetikos sistemoje.....	67
IŠVADOS .....	68
LITERATŪROS SĄRAŠAS .....	70
PRIEDAI.....	73

## PAVEIKSLŲ SĄRAŠAS

2.1.1 pav. Pilnutinė varža.....	14
2.1.2 pav. Gedimo vietos nustatymo pagrįsto varžų matavimu vienos pusės maitinimo linijose principinė schema.....	15
2.1.3 pav. b) varžos ir talpos reaktyvumas (vadinamas angl. reactance effect). c) varžos ir induktyvumo reaktyvumas.....	15
2.1.4 pav. Gedimo vietos nustatymo pagrįsto varžų matavimu dviejų pusių maitinimo linijose principinė schema.....	15
3.1.1 pav. Alytus 1 matavimų pereinamojo proceso grafikas (srovės ir įtampos).....	24
3.2.1 pav. Alytus matavimai 1. A,B,C fazių įtampų pereinamieji procesai gedimo metu.....	27
3.2.2 pav. Alytus matavimai 1. A,B,C fazių, srovių. Bei nulinės sekos srovės pereinamieji procesai gedimo metu.....	28
3.2.3 pav. Alytus matavimai 1. A,B,C fazių. Vidutinių reaktyviųjų varžų kitimo kreivės gedimo metu.....	28
3.2.4 pav. Alytus matavimai 1. A,B,C fazių. Vidutinių aktyviųjų varžų kitimo kreivės gedimo metu.....	29
3.2.5 pav. Alytus matavimai 1. A,B,C fazių. Pilnutinės varžos fazių kampų kitimo kreivės gedimo metu.....	29
3.2.6 pav. Alytus matavimai 1. A,B,C fazių. Vidutinių pilnutinių varžų kitimo kreivės gedimo metu.....	30
3.3.1 pav. Vidutinė pilnutinė varža, kai duomenys išfiltruoti su DC filtru ir be jų. Filto įtaka duomenims (Alytus 1 matavimas).....	33
3.3.2 pav. Vidutinė pilnutinė varža, kai duomenys išfiltruoti su DC filtru ir be jų. Filto įtaka varžos parinkimui (Alytus 1 matavimas).....	33
3.3.3 pav. Vidutinė aktyvioji varža, kai duomenys išfiltruoti su DC filtru ir be jų. Filto įtaka duomenims (Alytus 1 matavimas).....	34
3.3.4 pav. Vidutinė aktyvioji varža, kai duomenys išfiltruoti su DC filtru ir be jų. Filto įtaka varžos parinkimui (Alytus 1 matavimas).....	34
4.1 pav. Principinė modeliuojamo gedimo schema Simulink programoje.....	37
4.1.1 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos.....	40
4.1.2 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos.....	40

4.1.3 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras.....	43
4.1.4 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras.....	43
4.1.5 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras, bet nevertinama gedimo varža.....	46
4.1.6 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras, bet nevertinama gedimo varža.....	46
4.1.7 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos, kai varžos parenkamos pagal linijos kampą.....	49
4.1.8 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos, kai varžos parenkamos pagal linijos kampą.....	49
4.1.9 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos, kai varžos parenkamos pagal linijos kampą.Naudojant DC filtrą.....	52
4.1.10 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos, kai varžos parenkamos pagal linijos kampą.Naudojant DC filtrą.....	52
4.1.11 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras.Metodika 2.....	55
4.1.12 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras.Metodika 2.....	55
4.2.1 pav. Modeliuojamo gedimo fazė – fazė-žemė įtampos kitimo kreivė. Simulink programoje...57	
4.2.2 pav. Modeliuojamo gedimo fazė – fazė-žemė srovės kitimo kreivė. Simulink programoje.....57	
4.2.3 pav. Modeliuojamo gedimo fazė – fazė-žemė vidutinės reaktyviosios varžos gedimo metu kitimo kreivė.....	58
4.2.4 pav. Modeliuojamo gedimo fazė – fazė-žemė vidutinės pilnutinės varžos gedimo metu kitimo kreivė.....	58
4.2.5 pav. Modeliuojamo gedimo fazė – fazė-žemė vidutinės pilnutinės varžos gedimo metu kitimo kreivė.....	59
4.2.6 pav. Modeliuojamo gedimo fazė – fazė-žemė varžų fazių kampai gedimo metu.....	59
5.1.1 pav. Bandymų metu nustatytų, skaičiavimų tikslumo vidurkių grafikas.....	61
5.2.1 pav. Bandymų metu nustatytų skaičiavimų laiko trukmės (greičio) vidurkių grafikas.....	62
5.3.1 pav. Bandymų metu nustatytų skaičiavimų paklaidų vidurkių grafikas.....	64



## **LENTELIŲ SĄRAŠAS**

4.1. lentelė. Bandymo Nr.1 rezultatų lentelė.....	39
4.2. lentelė. Bandymo Nr.2 rezultatų lentelė.....	42
4.3. lentelė. Bandymo Nr.3 rezultatų lentelė.....	45
4.4. lentelė. Bandymo Nr.4 rezultatų lentelė.....	48
4.5. lentelė. Bandymo Nr.5 rezultatų lentelė.....	51
4.6. lentelė. Bandymo Nr.6 rezultatų lentelė.....	54

## SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS

DFRs – angl. digital fault recorders arba skaitmeniniai gedimų registratoriai;

FL – angl. fault locator arba gedimo lokatorius;

FIR – baigtinio impulso atsako filtras;

LSM – mažiausių kvadratų metodas;

DC – nuolatinė srovė;

$r$  – gedimo varža;

$f$  – sistemos dažnis;

$R$  – linijos aktyvioji varža;

$X_L$  – linijos reaktyvioji varža;

$Z$  – linijos pilnutinė varža;

$ZZ_{sk}$  – apskaičiuota algoritmo pilnutinė varža, mažiausia gedimo metu;

$RR_{sk}$  – apskaičiuota algoritmo aktyvioji varža, mažiausia gedimo metu;

$Z_{sk}$  – linijos pilnutinė varža;

AKI – automatinis kartotinis įjungimas.

## IVADAS

Pagrindinis dėmesys elektros energetikos perdavimo bei skirstomojo tinklo sistemose skiriamas elektros energijos perdavimo patikimumui bei kokybei. Nuolatos yra tobulinamos bei kuriamos perdavimo bei skirstomųjų tinklų valdymo sistemos. Jos privalo greitai bei efektyviai nustatyti tinklo trūkumus, gedimus bei juos pašalinti ir atkurti elektros energijos tiekimą vartotojams per kuo trumpesnę laiką. Greitam gedimo vietos nustatymui ir energijos tiekimo paslaugos atkūrimui naudojami įvairūs algoritmai bei metodai. Vienas iš sudėtingiausių uždavinių yra greitas gedimų vienoje elektros tinklo fazėje nustatymas bei atpažinimas. Kintantys apkrovos parametrai elektros tinkle, gedimo vietos nustatymo neapibrėžtumai, valdymo sistemų bei dispečerinio valdymo selektyvumas sukuria sutrikimus ir trigdžius, kartais tam tikrus apribojimus. O tai lemia netikslų, klaidingą gedimo vietos atpažinimą.

Dėl didelio matuojamų parametrų kiekio reikalingumo, šių parametrų tikslumo bei tikslaus ir greito perdavimo poreikio, tai pat dėl ilgai trunkančio duomenų apdorojimo, literatūroje nėra išsamiai išnagrinėtas algoritmų praktinis pritaikymas elektros energetikos sistemoje. Perdavimo tinklų sistemose gedimo vietos nustatymo ir atpažinimo uždavinių sprendime yra pasiektas gana aukštas lygis, kai gedimas yra vienoje tinklo fazėje. Tačiau skirstomajame tinkle ši problema nėra išsamiai išnagrinėta. Pasaulinėje literatūroje yra pasiūlyta nemažai įvairiausių gedimo vietos nustatymo algoritmų, tokių kaip gedimo vietos nustatymas paremtas registruotų parametrų analize ar technologijos paremtos tinklo pramoninio dažnio parametrų analize. Tačiau praktinis pritaikymas elektros energetikos sistemoje nėra nuodugnai ištirtas bei aprašytas. Todėl šią problemą reikia spręsti. Svarbu nustatyti kaip greitai bei tiksliai galima surasti gedimo vietą pasitelkiant algoritmus. Taip pat svarbu išsiaiškinti ar jie pritaikomi elektros energetikos sistemose. Remiantis šaltiniais [2,8,9,10,11,12,1,20,18,19,14,6] ir kitais naudojamais šiame darbe.

Šiame darbe sukūriau platformą, kuri padeda nustatyti atstumą iki gedimo vietos įvykus avarijai.

Tai padariau pritaikant gedimo vietos nustatymo algoritmus „MATLAB“ programa. Ištyriau kokie tikslūs šie algoritmai su mano pateikta platforma, bei kaip greitai jie atlieka skaičiavimus gedimo vietai surasti. Išbandžiau savo sukurtą modelį su realiais gedimo duomenimis. Taip pat sumodeliavau avarijas elektros energetikos sistemoje su Simulink simuliacine programa. Su gautais įmituotų gedimų duomenimis nustaciau algoritmų tikslumą bei jų greitį. Pagal tai įvertinau kokios yra šios platformos galimybės. Taip įvertinau šio modelio pritaikymą elektros energetikos sistemoje.

Pirmajame skyriuje trumpai apžvelgiau gedimo vietos nustatymo tikslą bei svarbą. Antrajame skyriuje pateikiau teorinę medžiagą apie naudojamus algoritmus bei metodikas. Trečiajame skyriuje apžvelgiau IEEE C37.111 – 2013 standarto panaudojimą bei pateikiau kuriamos modelio platformos

kodą. Trumpai aprašiau kas vyksta jį panaudojus. Taip pat pateikiau pritaikyto modelio duomenis su realiais duomenimis bei algoritmų kodus „MATLAB“ programoje, bei rezultatus gautus pritaikius algoritmus su realiais duomenimis. Taip pat pritaikomas nuolatinės srovės dedamosios šalinimo filtras su modeliu. Ketvirtajame skyriuje aprašau modeliavimą ir bandymus su Simulink progamos pagalba. Pateikiau 6 bandymus su skirtingais algoritmais. Taip pat ištyriau modelį su skirtingomis imituojamomis avarijomis. Penktajame skyriuje tyriau bei aprašiau šio modelio tikslo, skaičiavimo greičio bei paklaidos įvertinimus. Palyginau visus bandymus. Šeštajame skyriuje pateikiau šios platformos privalumus ir trūkumus. Įvertinau galimybę pritaikyti šį modelį elektros energetikos sistemoje.

**Darbo tikslas:** Magistrinio tiriamojo darbo tikslas yra išanalizuoti gedimo vietos nustatymo algoritmų metodikas pasaulinėje literatūroje. Sukurti praktinę panaudojimo platformą gedimo vietai nustatyti ir ištirti jos galimybes bei pritaikymą elektros energetikos sistemose.

#### **Uždaviniai:**

1. Išanalizuoti pasaulinėje literatūroje pateiktas gedimo vietos nustatymo algoritmų metodikas.
2. Ištirti kaip veikia gedimo vietos nustatymo algoritmai.
3. Sukurti gedimo vietos nustatymo algoritmo platformą, kurią galima pritaikyti praktiškai. Bei ištirti kokios jos galimybės.
4. Ištirti kaip tiksliai veikia sukurtas modelis su algoritmais bei juos palyginti tarpusavyje.
5. Ištirti kaip greitai veikia sukurta platforma su gedimo vietos nustatymo algoritmais.
6. Ištirti ar sukurta gedimo vietos nustatymo platformą galima pritaikyti elektros energetikos sistemose.

## 1. GEDIMO VIETOS NUSTATYMO TIKSLAS IR SVARBA

Gedimo vietos nustatymo tikslas nustatyti įvykusio gedimo vietą maksimaliai įmanomai tiksliai. Gedimo lokatoriai yra svarbiausi papildomos apsaugos įrenginiai, kurie pritaiko gedimo vietos nustatymo algoritmus. Jie apskaičiuoja atstumą iki gedimo vietos.

Kai gedimus nustatyti linijoje reikia daugiau nei vienoje sekcijoje, t.y. kai turime trijų terminalų arba daugiau terminalų liniją, sugedusi sekcija turi būti indentifikuota ir gedimo vieta šioje sekcijoje turi būti nustatyta.

Gedimo vietos nustatymo funkcijos gali būti realizuotos:

- mikroprocesoriuose – skirtiems apsauginėms relėms;
- skaitmeniniuose gedimų registratoriuose (DFRs);
- nešiojamuose gedimo vietos nustatymo prietaisuose;
- po gedimo panaudoti analizės programose.

Įtraukus gedimų nustatymo funkciją kaip papildomą mikroprocesorių funkciją, tai galima paprastai panaudoti praktikoje relinėje apsaugoje. Šiuo atveju didelis skaičiavimo pajėgumas ir ryšys su šiuolaikinių relių nuotolinėmis svetainėmis nereikalauja arba beveik nereikalauja jokių sąnaudų. Be to, skaitmeniniai gedimų įrašymo įtaisai leidžia lengvai ir nebrangiai įtraukti gedimų nustatymo funkciją. Savo ruožtu savarankiški gedimų lokatoriai taikomi naudojant sudėtingus gedimų nustatymo algoritmus su sąlyga, kad bus priimtos didesnės įgyvendinimo sąnaudos.

Kita galimybė yra susijusi su po gedimų (surinktų duomenų) analizės programomis, kuriose naudojami gedimo vietos nustatymo algoritmai. Tokios programos daugiausia naudojamos operacijų patikrinimui apsauginėse relėse.

Perdavimo ir skirstymo linijos patiria laikinus ir ilgalaikius gedimus. Laikini gedimai, kurie yra labiausiai dominuojantys oro linijų gedimai, dažniausiai savaimė pasišalina. Todėl elektros energijos tiekimo tęstinumas ilgai nebūna sutrikdytas, kas yra gerai. Esant ilgalaikiui gedimui relinės apsaugos įtaisas saugantis tą zoną kur įvyko gedimas siunčia signalą į jungtuvą (kuris yra arčiausiai gedimo vietos), kad jis atjungtų tą sekciją kurioje yra gedimas. Jei tam tikra linija neveikia, remontuojama ar joje įvyksta gedimas, tai šioje linijoje nebegali vykti elektros energijos tiekimas vartotojams. Kol gedimai bus pašalinti elektros vartotojus turės aprūpinti kitos veikiančios linijos.

Šiuolaikinės energijos sistemos veikia vis arčiau savo veiklos galimybių ribų. Todėl, kad būtų išvengta elektros energijos tiekimo nutraukimo („blackout“), ypatingą dėmesį reikia skirti ir negailėti sąnaudų įrengiant energetikos sistemas su apsaugos ir valdymo prietaisais, taip pat negailėti sąnaudų jų nustatymui (reguliavimui) bei kokybiškam valdymui. Pagal šaltinius [1,2,3,4,5].

## 2. GEDIMO VIETOS NUSTATYMO ALGORITMAI

### 2.1. Gedimo vietos nustatymas paremtas pilnutinės varžos matavimu

Pagrindinis parametras vienos pabaigos linijų algoritmuose yra sugedusios linijos varža, kuri padeda nustatyti gedimo vietą. Tarkime turime vienos fazės liniją (A – B), prijungtą prie šaltinio tik viename gale (A), t.y. linija kuri neturi apkrovos (B), kaip pavaizduota paveikslėlyje 2.1.1. Liniją paveikia gedimas (F). Kuris yra nutolęs nežinomu atstumu ( $d$ ) nuo mazgo (A) prie kurio yra įrengtas gedimo lokatorius (FL). Jei linijos A srovė nekinta, tai gedimo srovė  $I_F$  lygi  $I_A$  srovei. Varža, kurią apskaičiuoja gedimo lokatoriaus terminalas, panaudojant srovę  $I_A$  ir įtampą  $U_A$ , išreiškiama taip:

$$\underline{Z}_{FL} = \frac{U_A}{I_A} = d\underline{Z}_L + R_F \quad (2.1);$$

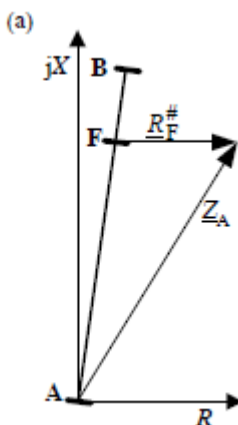
Atstumas iki gedimo apskaičiuojamas taip :

$$d = \frac{Z_{FL}}{Z_L} \quad (2.2);$$

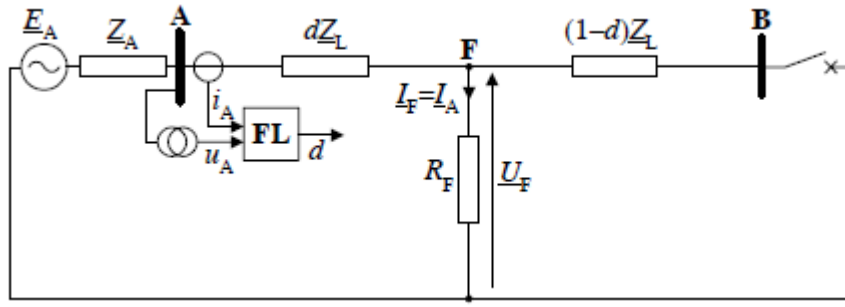
Šiomis formulėmis yra paremti vieno galo gedimo vietos nustatymo algoritmai, kurie aprašyti [5] šaltinyje.

Tai leidžia tiksliai nustatyti atstumą iki gedimo vietos vieno galo linijose. Tačiau tik tada kai gedimo lokatorius mato  $R_F$  kaip 2.1.1 pav. pavaizduotoje figūroje.

Čia pavaizduota pilnutinė varža bei aktyvioji varža (tikroji gedimo). Taip teigiama [2] šaltinyje.

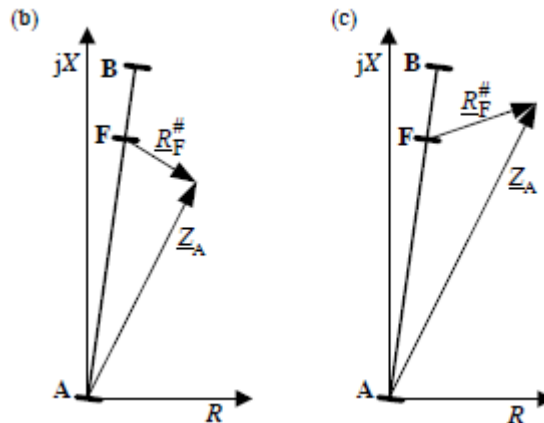


2.1.1 pav. Pilnutinė varža. [2]

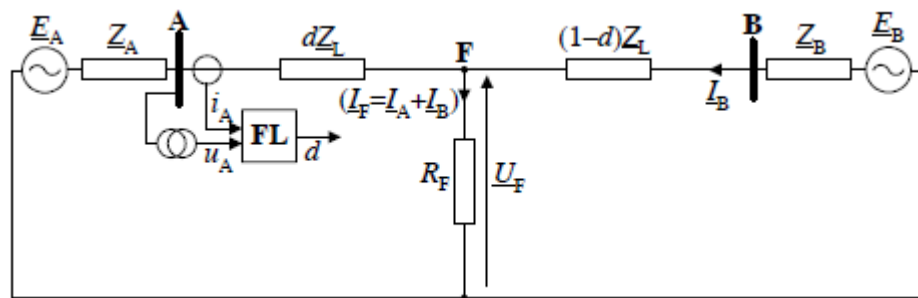


2.1.2 pav. Gedimo vietos nustatymo pagrįsto varžų matavimu, vienos pusės maitinimo linijose principinė schema.[2]

Jei yra dviejų maitinimų linija, kaip parodyta (2.1.4 pav.), gedimo srovė  $I_F$  nėra lygi srovei kuri yra gedimo lokatoriuje  $I_A$ , kol nuolatinė srovė  $I_B$  įeiną į gedimo srovės skaičiavimą ( $I_F = I_A + I_B$ ). Todėl kaip šalutinį rezultatą matomą gedimo lokatoriaus terminale gauname reaktyviosios varžos efektą kuris iškreipia varžas. Padidinama arba sumažinama pilnutinė varža. Dėl ko sunkiau nustatyti ar įvyko gedimas. Taip tegiama [2] šaltinyje. Bei tai parodyta 2.1.3 pav.



2.1.3 pav. b) varžos ir talpos reaktyvumas (vadinamas angl. reactance effect). c) varžos ir induktyvumo reaktyvumas (vadinamas angl. reactance effect).[2]



2.1.4 pav. Gedimo vietos nustatymo pagrįsto varžų matavimu dviejų pusių maitinimo linijose principinė schema.[2]

Čia pateiktas paprasčiausias būdas, kuris naudojamas nustatyti gedimo vietą, panaudojant varžų matavimus vieno maitinimo linijose.

## **2.2. Gedimo vietos nustatymas naudojant DC dedamųjų reikšmių pašalinimą**

Nors įtampos signalai praktiškai nėra užteršti, nykstančios nuolatinės srovės dedamosios srovės gali eksponentiškai mažinti DC tinklo balansą (dėl mažo kompensavimo) dėl nežinomų pradinių dydžių ir laiko konstantos.

Tai yra dėl to, kad gedimo metu esanti varža ir gedimo pradinis kampas yra atsitiktiniai dydžiai. Mažėjanti DC tinklo kompensacija esant gedimo srovei gali sukelti iki 20 procentų tinklo perkrovą (pagal skaičiavimus [2] šaltinyje, taip pat ten pateikta platesnė informacija apie šį algoritmą).

Gedimo vietos nustatyme DC dedamosios įtakoja distancijos iki gedimo paklaidą, ypač tada kai matavimų nėra daug.

Apskritai, fazorių vertintojai gali būti lengvai pritaikyti pašalinimui pasirinktai DC dedamajai, taip teigiama [8,9,10] šaltiniuose. Problema iškyla tada, kai nėra žinoma laiko konstanta DC tinklo balanse (kompensavime). Keletas pasiūlymų buvo pateikta kaip sumažinti DC dedamųjų įtaką fazorių vertinime. Pats praktiškiausias sprendimas sujungiant skaitmeninį imitacinį filtrą su bet koku tvirtu fazoriu yra naudojant Fourier pilno periodo algoritmą. Skaitmeninis imitacinis filtras skaitmeniniu būdu atkuria analoginę imitacinę grandinę, kuri dažniausiai naudojama tradicinių linijų apsaugai, taip teigiama [11, 8] šaltiniuose. Originali imitacinė grandinė tai diferencijuojantis elementas ir todėl jo skaitmeninė kopija yra aukšto lygio skaitmeniniai filtrai [12, 8] taip teigiama šiuose šaltiniuose. Tai savo ruožtu yra rezultatas blogėjančios dažnio charakteristikos, kurioje yra fazorių vertinimas. Didžiausias privalumas skaitmeninio imitacinio filtro yra tas, kad jis neiškraipo duomenų esant skirtingai laiko konstantai.

Kitas pasiūlymas yra gautų signalų modelyje pritakyti DC dedamųjų šalinimą (triukšmų šalinimą) ir kartu taikyti tam tikras kreivių braižymo technikas siekiant tikslingai suprojektuoti fazorių vertintoją. Paprastai DC kompensavimas laikomas triukšmu ir pateikiamas tikimybinėmis ir determinisinėmis charakteristikomis, o ne kaip parametras, kuris turi būti pritaikytas [12, 8] taip teigiama šiuose šaltiniuose. Šiuo pagrindu yra parašyta keletas specifinių vertinimų:

- Linijiniai įverčiai, kai naudojami norint sumažinti DC dedamąją. Yra susiję su netiesine eksponentinė funkcija naudojant keletą pirmųjų Taylor eilučių terminų.[9]
- Mažiausių kvadratų metodas (LSM) duoda nerekursinius ar rekursinius įvertinimus [11,13,14]. Pastarasis gali būti pakeistas įvedant atstatymo arba panaikinimo faktorius. Siekiant geriau įsisavinti neharmoninius dažnius įskaitant nykstančias DC dedamąsias.



- Būsenos kintamojo modelio taikymas suteikia pirminiams vertintojams būsenos stebėtojo pavidalą [15], kuris aprašytas šiame šaltinyje arba – Kalman filtro [16], kuris aprašytas šiame šaltinyje. Reikalavimai retai galimoms triukšmų charakteristikoms yra pagrindinis šių aplikacijų matavimų technikos trūkumas.

Dar vienas paprastas metodas pašalinti DC dedamąją yra pritaikyti kosinuso filtrą, kad įvertintume tiek tiesioginį, tiek kvadratinį stačiuosius komponentus, taip teigiama [2] šaltinyje. Papildomas uždelsimas lėtina vertinimo greitį ir atsakymo pateikimą. Tai yra šio metodo pagrindinis trūkumas.

Nors pirmiau minėti metodai pagerina matavimus, jie neidealiai atmeta DC komponentą. Žemiau pateiktas dar vienas algoritmas, kuris pilnai ir praktiškai idealiai pašalina DC dedamąją iš įeinančių signalų nepriklausomai nuo laiko konstantos ir pradinių dydžių.

Tolimesniam svarstymui priimame tokį signalo modelį kaip pateikta čia:

$$x(t) = X_0 \exp\left(-\frac{t}{\tau}\right) + \sum_{m=1}^K X_m \cos(m\omega_1 t - \varphi_m) \quad (2.3);$$

čia:

$X_0$ ,  $\tau$  – nykstančios DC dedamosios, laiko konstantos dydis;  $K$  – nagrinėjamų harmonikų skaičius;  $\omega_1 = 2\pi/T_1$  – radialinės sistemos dažnis;  $T_1$  – fundamentalaus dažnio periodas;  $\varphi_m, Y_m$  – fazės kampo reikšmė ir  $m$  – tosios harmonikos reikšmė.

Kad išskirti fazorių iš  $M$  skaičiaus nuoseklių signalų pateiktą formulėje (2.3) mes turime naudoti LSM algoritmą kartu su modelio signalų matrica  $\mathbf{H}(k)$ , pagal [17] šaltinį.

Ši matrica sudaroma iš toliau darbe aprašomų išraiškų:

$$\mathbf{h}(k-j) = [h_R(k-j), h_I(k-j), h_0(k-j)], j = 0 \dots M-1 \quad (2.4);$$

čia,

$$h_R(k-j) = \cos(\mathbf{v}(k-j)), h_I(k-j) = -\sin(\mathbf{v}(k-j))$$

$$h_0(k-j) = \exp(b(k-j))$$

kur:

$$\mathbf{v} = \frac{2\pi}{N}, b = -\frac{T_1}{N\tau}, N - \text{skaičius pavyzdžių per ciklą. } k - \text{pavyzdžio indeksas.}$$

Apsvarstykime viso periodo ( $M=N$ ) standartinį LSM algoritmą paremtą signalo modelio matrica (2.4 formulė) kartu su žinoma laiko konstanta  $\tau$ . Skaičiavimų vektoriai yra atnaujinami su kiekvienu skaičiavimu, nuolatos naudojant fundamentalias išraiškas kurios pateiktos toliau:

$$\hat{\mathbf{X}}(k) = \mathbf{P}(k)\mathbf{H}^T(k)\mathbf{x}(k) \quad (2.5);$$

kur, kovariacijos matrica:

$$\mathbf{P} = (\mathbf{H}^T(k)\mathbf{H}(k))^{-1}$$

ir

$$\mathbf{x}(k) = [x(k - N + 1), x(k - N + 2), \dots, x(k)]^T, \quad \hat{\mathbf{X}}(k) = [X_R(k), X_I(k), X_0(k)]^T$$

kol yra viršutinis matricos indeksas T.

Praktiškai tik du kintamieji  $X_R(k)$  ir  $X_I(k)$  turi būti nustatomi formule (2.4). Kol jie formuoja apskaičiuotus fazorius.

$$u(t) = Ri(t) + L \frac{di(t)}{dt} + \frac{1}{C} \int_0^t i(\tau) d\tau \quad (2.6);$$

Algoritmas (2.5) atmeta DC dedamąją idealiai tol kol įeinantis signalas atitinka tariamą signalo modelį (2.6). Standartinis LSM algoritmas yra labai jautrus laiko konstantai  $\tau$  ir negali būti panaudotas tiesiogiai skaitmeninėse relėse, taip teigiama [8] šaltinyje.

Dėl eksponentiškai mažėjančios DC dedamosios buvimo signalo modelyje (2.5) matrica  $\mathbf{P}(k)$  formulėje (2.5) yra pilna matrica su išvesta įstrižaine elementų priklausomų nuo eksponentinės funkcijos. To pasekmėje, filtrai gaunami nerekursinio vertinimo metu, turi nereguliuos koeficientus ir jie negali būti pateikti rekursine forma kitaip nei pilno periodo Fourier algoritme.

Rekursinis algoritmas gali būti pritaikytas tam tikrai ypatingai LSM vertinimo formai, kaip pateikiama žemiau.

Įvertinkime svarbą LSM įvertinimo priemonės kartu su matavimo modelio matrica apibrėžta formulėje (4.15) pateiktoje [2] šaltinyje kartu su išraiškomis  $\mathbf{H}_G(k)$  pateiktomis taip:

$$\mathbf{h}_G(k - j) = [h_c(k - j), h_s(k - j), h_0(k - j)], j = 0 \dots N - 1 \quad (2.7);$$

čia,

$$h_c(k - j) = \cos(\mathbf{v}(k - j)) - d_c(k), \quad h_s(k - j) = -\sin(\mathbf{v}(k - j)) - d_s(k)$$

Verta pabrėžti, kad matrica  $\mathbf{H}_G(k)$  yra gaunama pagal formulę (2.7) ir tinkamai modifikuojant originalią signalo modelio matricą  $\mathbf{H}(k)$  (2.4 formulė). Šiai operacijai nebūtinai reikia žinoti svorių matricą  $\mathbf{G}(k)$ . Tačiau turi būti surastos adekvačios koeficientų  $d_c(k)$  ir  $d_s(k)$  vertės.

Matavimų modelio skaičiuoklis yra surandamas pagal (4.88) formulę pateiktą [2] šaltinyje, kartu naudojant sekančią kovariacijos matricą pateiktą čia:

$$\mathbf{P}_G(k) = (\mathbf{H}_G^T(k)\mathbf{H}(k))^{-1} \quad (2.8);$$

Funkcijos  $d_c(k)$  ir  $d_s(k)$  formulėje (2.7) yra fiksuotos signalo informacijos lange.

Kuriant vertinimą (2.5) šioms funkcijoms išskiriami du reikalavimai:

- kovariacijos matrica  $\mathbf{P}_G(k)$  galutiniame algoritme turi būti diagonalinė;
- Funkcijos  $d_c(k)$  ir  $d_s(k)$  turi būti lengvai surandamos iš DC dedamosios įėjimo signalo.

Pirmo reikalavimo tikslas yra surasti paprasčiausią galutinio matavimų algoritmo formą.

Tiesą sakant, pakanka įvesti šį reikalavimą tik pirmosioms dviem matricos  $\mathbf{P}_G(k)$  eilutėms, kurios yra susijusios su apskaičiuotomis stačiosiomis sudedamosiomis dalimis. Galiausiai pirmasis reikalavimas gali būti užrašytas taip:

$$q_{ij} = 0 \text{ kai } i \neq j, i = 1,2 \text{ } j = 1,2,3 \quad (2.9);$$

čia,

$$\{q_{ij}\} = \mathbf{H}_G^T(k)\mathbf{H}(k)$$

Iš šios sąlygos mes galime gauti šiuos santykius apibrėžiančias funkcijas  $d_c(k)$  ir  $d_s(k)$ , pagal šaltinį [17], pateiktas čia:

$$d_e(k) = \frac{\sum_{j=0}^{N-1} \cos(v(k-j))}{\exp(bk) \sum_{j=0}^{N-1} \exp(-bj)} \quad (2.10);$$

$$d_s(k) = \frac{-\sum_{j=0}^{N-1} \sin(v(k-j))}{\exp(bk) \sum_{j=0}^{N-1} \exp(-bj)} \quad (2.11);$$

kurios supaprastintos atrodo taip:

$$\begin{aligned} d_e(k) &= D \cos(vk + \vartheta) \\ d_s(k) &= -D \sin(vk + \vartheta) \end{aligned} \quad (2.12);$$

čia,

$$\vartheta = \tan^{-1}\left(\frac{\sin v}{\cos v - \exp(b)}\right)$$

$$D = \frac{1 - \exp(b)}{\sqrt{(1 - \exp(b))^2 + \sin^2 v}}$$

Diagonaliniai elementai pirmosiose dvejose eilutėse matricos  $\mathbf{P}_c(k)$  yra konstantos:  $1/q_{11} = 1/q_{22} = 2/N$ .

Iš formulės (2.12) aiškiai matosi, kad gautos funkcijos  $d_c(k)$  ir  $d_s(k)$  yra tiesiogiai skirtos DC komponento laiko konstantai. Tai atitinka antrojo reikalavimo sąlygas šioms dvejoms funkcijoms.

Analogiškai formulei (4.96), pateiktai [2] šaltinyje, mes parašome algoritmą skirtą fazorių vertinimui iš (2.5) ir (2.7) formulių, tai pateikta sekančiai:

$$X_e(k) = \frac{2}{N} \sum_{j=0}^{N-1} h_e(k-j)x(k-j) = X_R(k) - \delta_e(k) \quad (2.13);$$

$$X_s(k) = \frac{2}{N} \sum_{j=0}^{N-1} h_s(k-j)x(k-j) = X_I(k) + \delta_s(k) \quad (2.14);$$

čia,

$$X_R(k) = \frac{2}{N} \sum_{j=0}^{N-1} h_R(k-j)x(k-j) \quad (2.15);$$

$$X_I(k) = \frac{2}{N} \sum_{j=0}^{N-1} h_I(k-j)x(k-j) \quad (2.16);$$

$$\delta_e(k) = d_e(k)X_a(k)$$

$$\delta_s(k) = d_s(k)X_a(k) \quad (2.17);$$

$$X_a(k) = \frac{2}{N} \sum_{j=0}^{N-1} x(k-j) \quad (2.18);$$

Iš formulių (2.13) ir (2.14) galime matyti, kad  $X_R(k)$ ,  $X_I(k)$  yra originaliai stačiakampiai komponentai ir surandami naudojant reguliarų pilno periodo Fourier algoritmą. Naujos koreguotos stačiakampių komponentų  $X_e(k)$ ,  $X_s(k)$  reikšmės yra gaunamos naudojant tinkamą korekciją. Ji užrašyta formulėje (2.17). Koreguotos funkcijos  $\delta_e(k)$ ,  $\delta_s(k)$  yra vidutiniškai praporingos DC dedamajai duomenų lange  $X_a$  (2.18 formulė) ir funkcijoms  $d_e(k)$ ,  $d_s(k)$  formulė (2.17). Šis algoritmas gali būti perrašytas rekursine forma, tai aprašyta [18] šaltinyje.

Norint surasti nykstančią DC dedamąją šiam algoritmui vis tiek reikia laiko konstantos  $\tau$ . Koreliacijos funkcijos dabar išskirtos į dvi vertinimo modifikacijas. Tokia vertinimo forma leidžia surasti vertę  $\tau$ , ieškant fazorio reikšmės.

Dabar galime supaprastinti funkcijas  $d_e(k)$ ,  $d_s(k)$ . Iš formulės (2.12) gauname tokią išraišką:

$$d_e(k) = p_e h_R(k) + p_s h_I(k)$$

$$d_s(k) = p_s h_R(k) - p_e h_I(k) \quad (2.19);$$

čia,

$$\begin{aligned} p_e &= D \cos \vartheta, \\ p_s &= D \sin \vartheta. \end{aligned}$$

Koeficientai  $p_c$  ir  $p_s$  priklauso nuo konstantos  $v$  ir parametro  $r = \exp(b)$ . Parametras  $r$  gali būti surastas iš matavimų. Komponentas  $X_a(k)$ , išreikštas (2.18) formulėje, reiškia sumą geometrinės progresijos  $N$  elementų su daugikliu  $r = \exp(b)$ . Tai išreiškiami čia:

$$r = r(k) = \exp(b) = \frac{X_a(k)}{X_a(k-1)} \quad (2.20);$$

Vertinimo išraiška pateikta (2.20) formulėje yra pagrindinis šiame skyriuje aprašomo algoritmo elementas, nes tai leidžia tiksliai ir tiesiogiai įvertinti nežinomą laiko konstantą naikinant DC dedamąją. Taip teigiama [2] šaltinyje.

Fiziškai laiko konstanta  $\tau$  nepasikeičia per tam tikrą trumpą laiką. Vertintojas (2.20 formulė) gali pateikti nerealistiškus dydžius gautus per pirmus skaičiavimus nuo gedimo pradžios. Todėl šiam vertinimui reikia tam tikro priverstinio stabilizavimo. Tai galima padaryti tikrinant šias sąlygas:

$$r_{min} \leq r \leq r_{max} \quad (2.21);$$

taip pat, tai apriboja apimtį jei ribos viršijamos. Ribos yra natūraliai parenkamos taip:

$$r_{min} = \exp\left(\frac{-T_1}{N\tau_{min}}\right), r_{max} = 1 \quad (2.22);$$

čia,  $\tau_{min}$  yra minimali laiko konstantos vertė. Rekomenduojama, kad ji būtų 4 ms.

Tolesnis apskaičiuoto faktoriaus  $r$  nepageidaujamų svyravimų slopinimas gali būti atliktas vidutiniškai per tam tikrą bandymų skaičių  $L$ , naudojant tokią išraišką:

$$r_{filtruota}(k) = \frac{1}{L} \sum_{m=0}^{L-1} r_{originatus}(k-m) \quad (2.23);$$

Kaip rodo simuliacijos, geri rezultatai gaunami kai  $L = 2..4$ . tai pateikta šaltinyje [2]. Praktiškai tai yra nieko verta, jei filtras (2.23 formulė) neuždelsia fazoriaus įvertinimo. Tai yra todėl, kad formulė (2.23) taikoma signalo parametrai, kas fiziškai nepakeičia laiko.

Norint supaprastinti išraiškas naujam algoritmui koeficientus  $p_c$  ir  $p_s$  išreiškime kaip funkciją  $v$  ir  $r$ . Pakeičiant formulę (2.12) į (2.19) formulę ir pridėdant išraiškas  $\vartheta$  (2.12 formulė) ir (2.20 formulė), gauname tokią išraišką:

$$p_c = \frac{(1-r)(\cos v - r)}{(1-r)^2 + \sin^2 v}, \quad p_s = \frac{(1-r)\sin v}{(1-r)^2 + \sin^2 v} \quad (2.24);$$

Realaus laiko skaičiavimai gali būti apibendrinti taip:

1. Stačiakampius komponentus  $X_R(k), X_I(k)$  skaičiuoti naudojant formules (2.15) ir (2.16) t.y. naudojant pilno periodo Fourier algoritmą.  $X_a(k)$  skaičiuoti naudojant formulę (2.18).
2. Jei  $X_a(k-1) < \xi$  ( $\xi$  – minimali reikšmė DC dedamosios kuri yra nagrinėjama koregavime) naudoti  $X_R(k), X_I(k)$  kaip galutinį rezultatą ir praleisti 3 ir 4 žingsnius.
3. Nustatyti funkcijas  $d_c(k), d_s(k)$  vadovaujantis šia seka:
  - surasti parametras  $r$  naudojantis formulėmis (2.20) ir (2.21);
  - apskaičiuoti vertes  $p_c$  ir  $p_s$  naudojant formulę (2.24) arba iš atmintyje išsaugotų funkcijų (naudojant grafikus).
  - surasti  $d_c(k), d_s(k)$  pagal formulę (2.19).
4. Apskaičiuoti korekcijos koeficientus  $\delta_e(k)$  ir  $\delta_s(k)$  taip pat kaip ir galutines ortogonalinių (stačiakampių) komponentų vertes, pagal formules (2.13) ir (2.14).

Antras apibendrinimas užtikrina, kad dalinimo iš nulio nebus. Sąlyga taip pat negalioja korekcijai kai DC dedamoji yra maža ir pataisymas iš tikrųjų nenaudingas.

Pusiasvyros algoritmo našumas praktiškai toks pat kaip ir standartinio pilno periodo Fourier filtro.[2].

Yra galimybė sintezuoti šį algoritmą pusei ciklo, tačiau jis būtų gerokai jautresnis aukštesnėms harmonikoms įėjimo srovėje.[17] Plačiau apie tai pateikta šiame šaltinyje.

### **3. GEDIMO VIETOS NUSTATYMAS PAGAL LINIJOS ĮTAMPŲ IR SROVIŲ PEREINAMUOSIUS PROCESUS GEDIMO METU**

Šiame skyriuje aprašytas tyrimas, kurio metu pagal įtampos ir srovės kreives linijoje buvo nustatinėjama gedimo vieta. Tyrimas atliktas naudojant Lietuvos respublikos aukštos įtampos perdavimo tinklų realius duomenis (110 – 330 kV). Skaičiavimai atlikti „MATLAB“ programos pagalba.

#### **3.1. „COMTRADE“ standarto pritaikymas „MATLAB“ programoje**

Norint tikslingai atlikti tyrimą gedimo vietos nustatymui perdavimo linijoje, pirmiausia reikia apibrėžti kaip bus panaudojami duomenys surinkti iš skaitmeninių relių. Įvykus gedimui ar jam vykstant daugybė linijos duomenų yra fiksuojami ir išsaugomi skaitmeninėse relėse. Naudojant relių mikroprocesorius bei kitą programinę įrangą šie duomenys yra perkeliama į asmeninius kompiuterius ar kitą duomenų saugojimo platformą pvz. USB, CD diską.

„Comtrade“ standartas apibrėžia kokius duomenis turime gauti iš relės, kad būtų galima panaudoti juos tikslingai ir išanalizuoti tarp skirtingų individų. Šis standartas naudojamas tarp relinę apsaugą eksploatuojančių, įrenginėjančių ar analizuojančių specialistų. Todėl duomenys turi būti vertinami pagal tuos pačius IEEE C37.111 – 2013 standarto reikalavimus (pagal šaltinį [20]). Taigi iš relių privalomai turime gauti duomenis apie visų trijų fazių srovės ir įtampas bei nulinės sekos srovę. Taip pat pagal šį standartą relė turi pateikti duomenis apie gedimo laiką, bei informaciją kurioje linijoje jis įvyko. Turi būti pateikiami užfiksuoti įvykiai gedimo metu ir kas tada vyko linijoje pvz: atjungtas jungtuvas ir t.t. Turi būti žinomas analoginių bei diskretinių signalų kiekis. Turi būti pateiktas tinklo dažnis bei duomenų kiekis. Visi šie duomenys yra aprašyti dviejuose failuose. Pirmas failas – tai duomenys dat. faile, kuriame yra užfiksuotos visos reikšmės linijoje gedimo metu. Kitas failas yra cfg. failas, kuriame aprašyti „data“ duomenys pagal „COMTRADE“ standartą.

Pirmiausia atliekant tyrimą turėjau panaudoti duomenis pagal „COMETRADE“ standartą ir juos užkrauti bei apdoroti su „MATLAB“ programos pagalba tam, kad galėčiau atlikti skaičiavimus ir juos analizuoti. Taip tiriant duomenis galima rezultatus lyginti su kitų specialistų rezultatais, nes duomenys naudojami pagal vienodus reikalavimus. Platforma sukurta pagal šaltinius [20,6,7].

Priede Nr. 1 pateiktas kodas „MATLAB“ progamoje kaip aš apdorojau ir užkroviau duomenis pagal IEEE C37.111 – 2013 standartą. Po komentaro ženklų, kuris „MATLAB“ progamoje žymimas procento ženklu „%“, aprašyta kas atliekama naudojant komandas. Žr. priedas Nr. 1.

Taip pat šis kodas atvaizduoja srovių bei itampos fazių pereinamuosiu procesus (3.1.1 pav.), bei pateikia kokie tai matavimai buvo ir kada šis gedimas įvyko. Čia pateikiamas vieno iš realaus duomenų matavimo panaudojimo grafikas, kurį atvaizduoja Priede Nr. 1 pateiktas kodas, kai gedimas

įvyksta C fazėje su žeme. Pilnas kodas trumpai aprašytas ir kokios užduotys atliekamos pateiktos po komentaro ženklų. Žr. priedas Nr.1.

Čia pateikiami duomenys apie gedimą įvykusį Alytus – Seirijai 110 kV linijoje. Linijos ilgis 46,51 km. Gedimas: faze C – žemė.

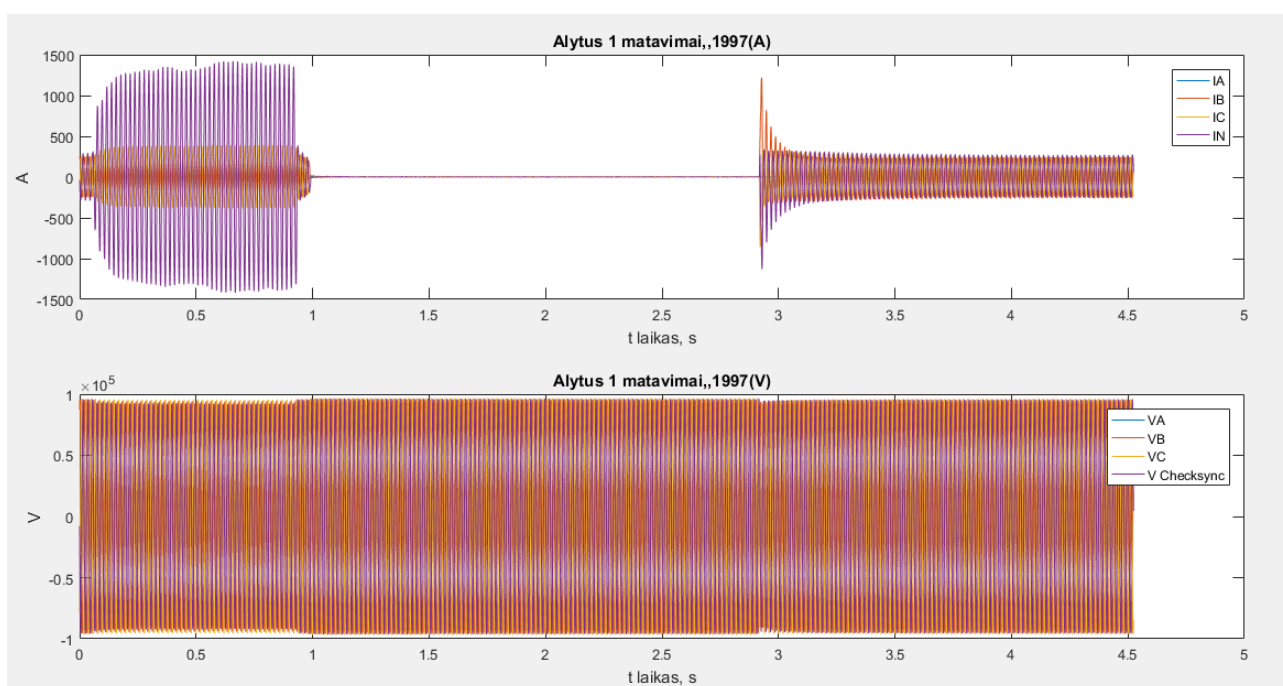
Toliau pateikta koda atvaizduoja šis kodas:

Duomenys: Alytus 1 matavimai

```
'ComtradeFile' 'Alytus 1 matavimai' 'ASCII'
```

```
'Daznis' '50.0' 'Hz'
```

```
Laikas = 11/07/2017,18:53:56.823000
```



3.1.1 pav. Alytus 1 matavimų

Šiuos realius duomenis pritaikysiu kartu su savo sukurtu skaičiavimo modeliu sekančiame skyriuje ir pateiksiu veiksmus, kurie yra atliekami su jais. Bei, kaip surandama gedimo vieta esant gedimui. Kodas sukurtas naudojantis šaltiniais [20,6,7].



### 3.2. Gedimo vietos nustatymas „MATLAB“ programos pagalba

Šiame skyriuje aprašomas gedimo vietos nustatymas panaudojant „MATLAB“ programą. Pateikiamas kodas, kuris suranda gedimo vietą, suskaičiuoja linijos vidutines varžas: aktyviają, reaktyviają ir pilnutinę. Bei nubraižo grafikus kaip kinta varžos skaičiuojamuoju laiko momentu. Apskaičiuojami pilnutinės linijos laidumas  $Y$ , taip pat aktyvusis laidumas  $G$ , bei reaktyvusis laidumas  $B_L$ . Apskaičiuojami varžų kampai ir jie atvaizduojami grafike. Atvaizduojamas visų fazių srovių kreivių kitimas laike. Taip pat visų fazių įtampų kitimas laike. Suskaičiuojama vidutinė pilnutinė varža ir ji atvaizduojama. Suskaičiuojamas avarijos laikas, bei AKĮ pauzė. Apskaičiuojamos mažiausia laidumo ir didžiausia laidumo vertės, bei surandamos vidutinės varžos. Pritaikomos skirtingos metodikos surasti gedimo vietą. Taip pat atliekama papildomi skaičiavimai. Jie pateikti pilname kode.

Priede pateikiamas pilnas kodas, bei komentaruose pažymėta kokie skaičiavimai ir kokie veiksmai atliekami tai pritaikius „MATLAB“ programą (kode matomi visi veiksmai, kurie atliekami „MATLAB“ programoje). Kodas sukurtas remiantis metodikomis [20,6,7,2,3,4,19,1] bei kt., ir pateiktas prieduose. Žr. Priedas Nr.2.

Toliau pritaikius skaičiavimus naudojamus šiame kode gaunu šiuos rezultatus su realiais duomenimis:

1.) Kai gedimo vieta skaičiuojama pagal formules aprašytas 4.1.1 skyriuje (4.7 formulė gedimo vietai rasti) ir ieškomas mažiausios varžų reikšmės  $C$  fazės gedimo metu. Reikšmės išrenkamos iš atvaizduotų vidutinių aktyviosios ir pilnutinės varžų reikšmių kitimo kreivių pateiktų žemiau (3.2.4 pav. Ir 3.2.6 pav.). Čia atsakymas lygus atstumui iki gedimo vietos 1:1.

$$Z_{sk} = 20,75, \Omega.$$

$$RR_{sk} = 37,2158, \Omega.$$

$$ZZ_{sk} = 75,1695, \Omega.$$

$$kint = \frac{ZZ_{sk} - RR_{sk}}{Z_{sk}} = \frac{75,1695 - 37,2158}{20,75} = 1,8291, Km.$$

čia,

$kint$  – atstumas iki gedimo vietos, km.

$ZZ_{sk}$  – apskaičiuota algoritmo pilnutinė varža mažiausia gedimo metu,  $\Omega$ .

$RR_{sk}$  – apskaičiuota algoritmo aktyvioji varža mažiausia gedimo metu,  $\Omega$ .

$Z_{sk}$  – linijos pilnutinė varža,  $\Omega$ .

2.) Kai gedimo vieta skaičiuojama pagal formulę (4.11). Ieškoma mažiausia pilnutinės varžos reikšmė  $C$  fazės gedimo metu. Reikšmės išrenkamos iš atvaizduotų vidutinių pilnutinės varžos reikšmių kitimo kreivių (3.2.6 pav.). Čia atsakymas lygus atstumui iki gedimo vietos 1:1.

$$Z_{sk} = 20,75, \Omega.$$

$$ZZ_{sk} = 75,1695, \Omega.$$

$$kint = \frac{ZZ_{sk}}{Z_{sk}} = \frac{75,1695}{20,75} = 3,6226, Km.$$

čia,

*kint* – atstumas iki gedimo vietos, km.

3.) Kai skaičiuojama pagal tokias formules (4.7 formulė gedimo vietai rasti), yra ieškoma aktyvioji ir pilnutinė varžos pagal linijos kampą kuris yra 64 laipsniai. Tada išrenkamos mažiausios reikšmės tuo pačiu momentu, kai yra toks linijos kampas. Čia atsakymas lygus atstumui iki gedimo vietos 1:1.

Rezultatai gaunami tokie:

$$Z_{sk} = 20,75, \Omega.$$

$$RR_{sk} = 41,2351, \Omega.$$

$$ZZ_{sk} = 94,0644, \Omega.$$

$$kint = \frac{ZZ_{sk} - RR_{sk}}{Z_{sk}} = \frac{94,0644 - 41,2351}{20,75} = 2,5460, Km.$$

čia,

*kint* – atstumas iki gedimo vietos, km.

4.) Kai skaičiuojama gedimo vieta pagal tokią formulę (4.11) pilnutinė varža imama pagal linijos kampą kuris yra 64 laipsniai. Tada ieškoma mažiausios jos reikšmės, pagal parinktą linijos kampą.

Rezultatai gaunami tokie:

$$Z_{sk} = 20,75, \Omega.$$

$$ZZ_{sk} = 94,0644, \Omega.$$

$$kint = \frac{ZZ_{sk}}{Z_{sk}} = \frac{94,0644}{20,75} = 4,5332, Km.$$

čia,

*kint* – atstumas iki gedimo vietos, km.

Taip pat naudojamas „MATLAB“ kodas išskaičiuoja avarijos laiką, kiek truko avarija, AKI (automatinis kartotinis įjungimas) laiką bei kiek tai truko.

Gedimo metu rezultatai gaunami tokie:

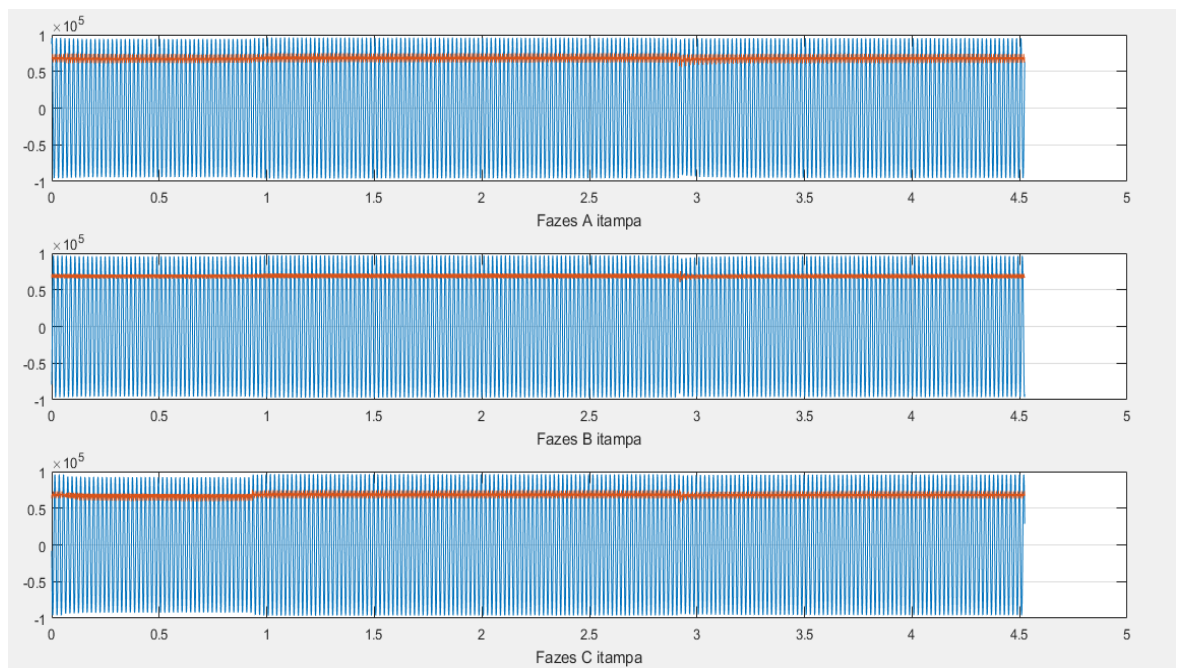
$$T_{avarijos} = 0.0438 \quad 0.9326, \text{ čia gedimo laikas s.}$$

$$dT_{avarijos} = 0.8888, \text{ čia gedimo trukmė, s.}$$

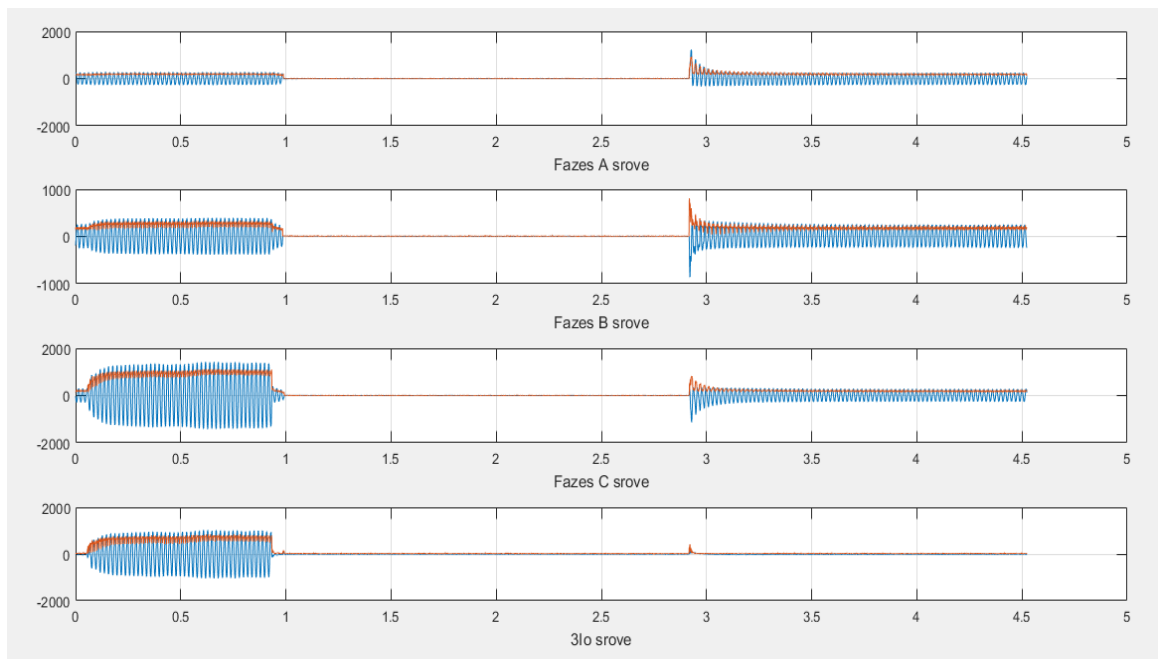
Taki = 0.9339 2.9021, čia AKĮ laikas ,s.

dTaki = 1.9682, čia AKĮ trukmė, s.

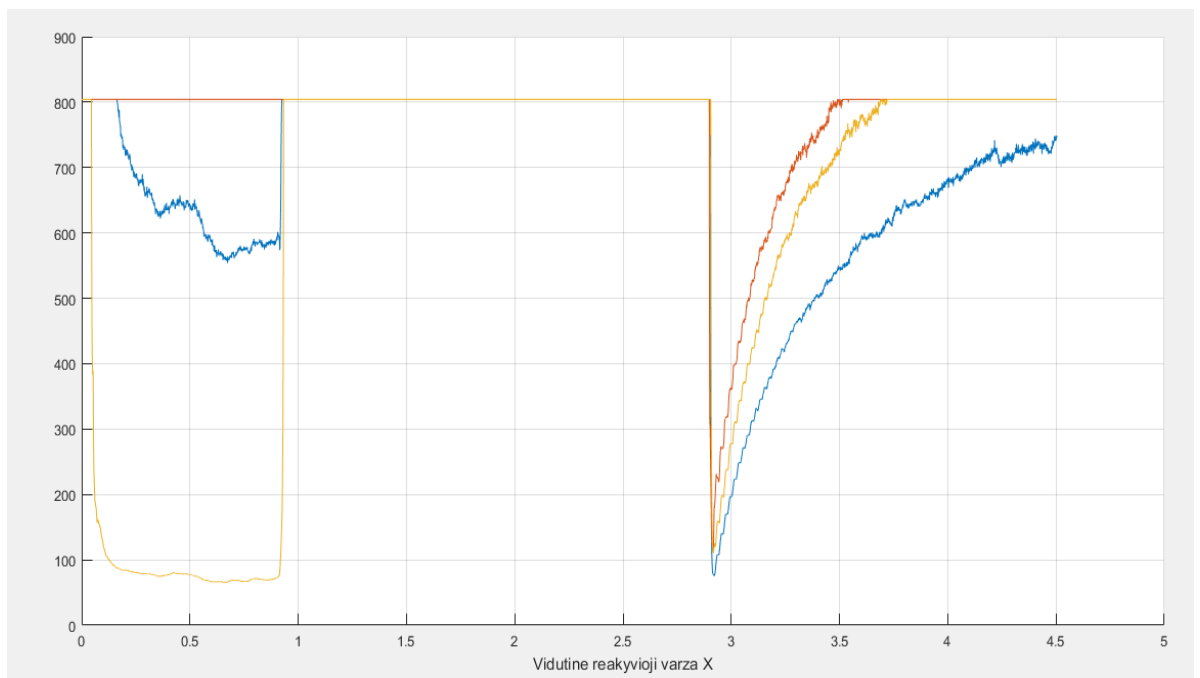
Toliau pateikiami realių duomenų, apskaičiuotų pagal šį modelį, atvaizduoti visų fazių srovių, įtampų pereinamieji procesai. Taip pat apskaičiuotos ir atvaizduotos gedimo metu esančios vidutinės aktyvioji, reaktyvioji ir pilnutinė varžos linijoje. Parodytos vidutinės pilnutinės varžos kampų kitimo kreivės gedimo metu. Šiuos duomenis išskaičiuoja mano pateikti „MATLAB“ programos kodai. Naudojami kodai. Žr. priedas 1 ir priedas 2.



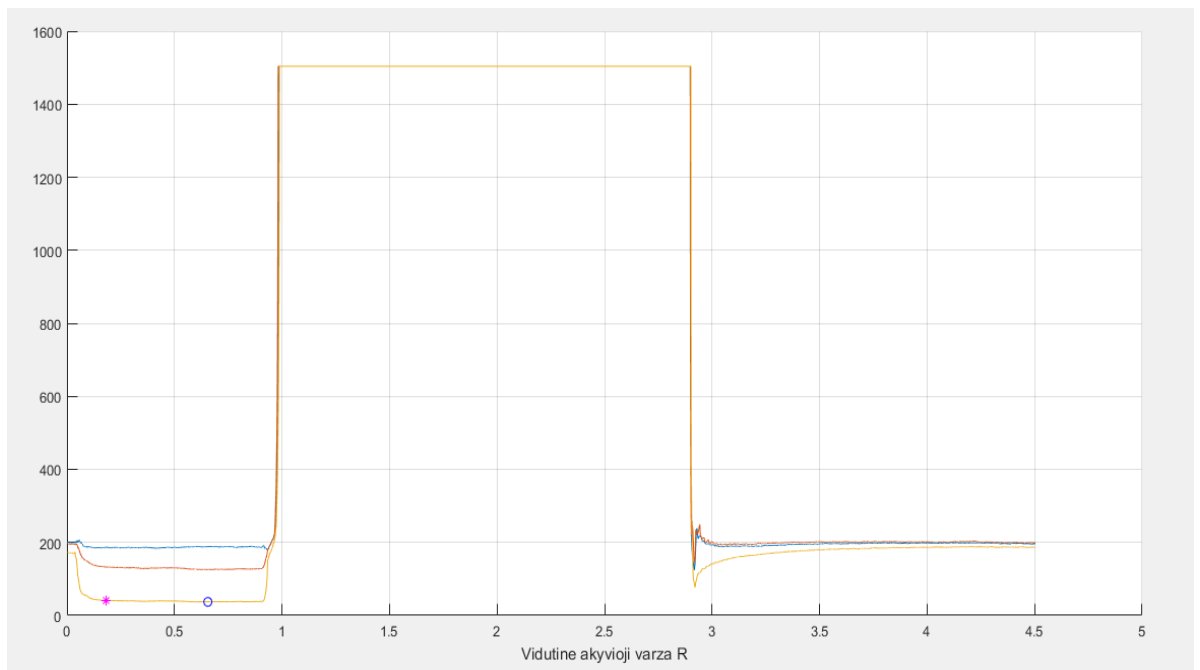
3.2.1 pav. Alytus matavimai 1. A,B,C, fazių įtampų pereinamieji procesai gedimo metu.



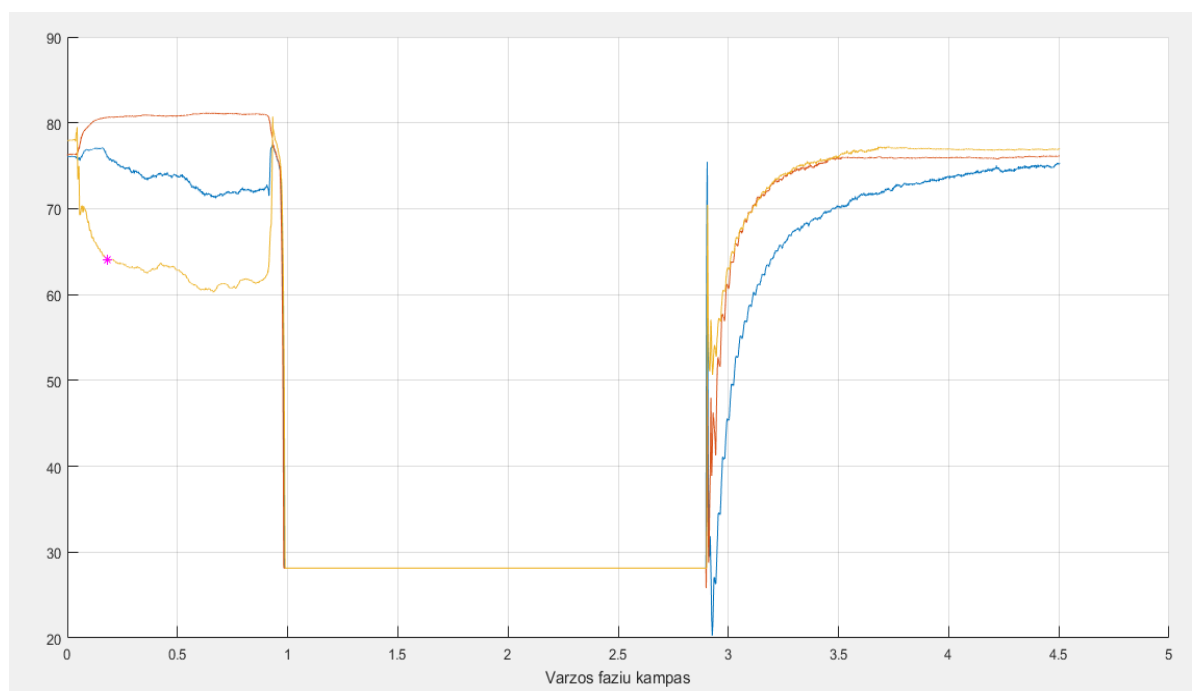
3.2.2 pav. Alytus matavimai 1. A,B,C fazių, srovių. Bei nulinės sekos srovės pereinamieji procesai gedimo metu.



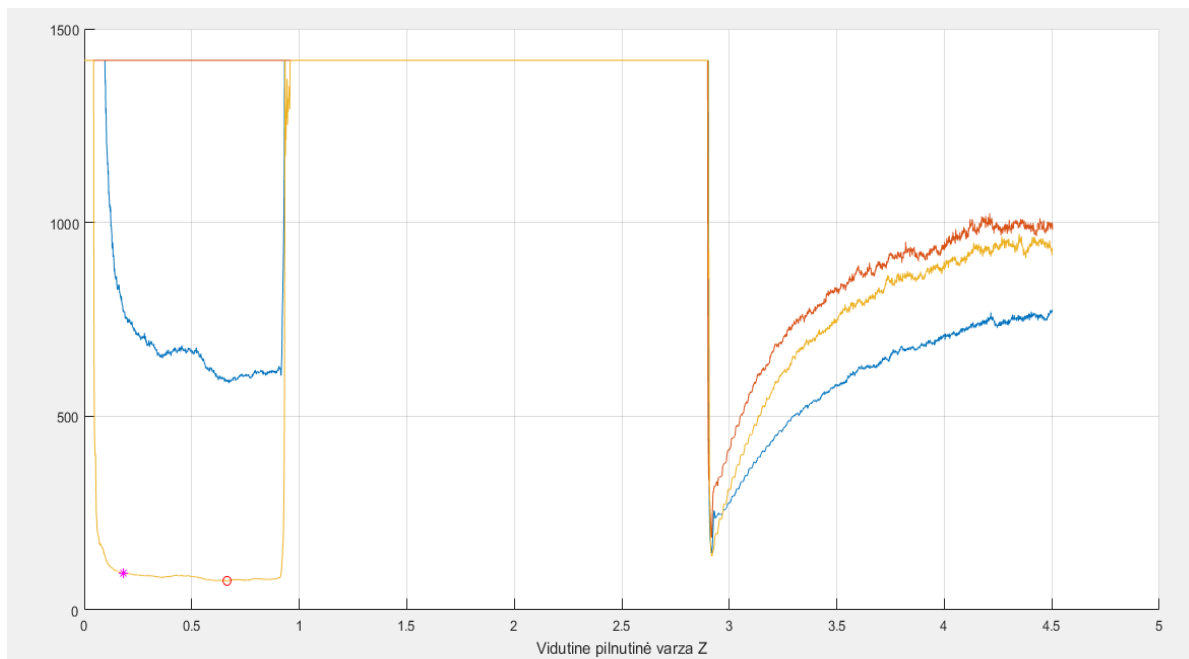
3.2.3 pav. Alytus matavimai 1. A,B,C fazių. Vidutinių reaktyviųjų varžų kitimo kreivės gedimo metu.



3.2.4 pav. Alytus matavimai 1. A,B,C fazių. Vidutinių aktyviųjų varžų kitimo kreivės gedimo metu.



3.2.5 pav. Alytus matavimai 1. A,B,C fazių. Pilnutinės varžos fazių kampų kitimo kreivės gedimo metu.



3.2.6 pav. Alytus matavimai 1. A,B,C fazių. Vidutinių pilnutinių varžų kitimo kreivės gedimo metu.

Šiuose grafikuose žymėjimas toks: A fazė – mėlyna spalva. B fazė – oranžinė spalva. C fazė – geltona spalva (3.2.1 – 3.2.6 pav.). Įtampos ir srovių kreivėse mėlyna spalva žymima įtampa ar srovė, o oranžine spalva įtampos ar srovės vidutinės išvestinės (priklausomai nuo grafiko 3.2.1 pav. ir 3.2.2 pav.). Žvaigždutė žymi kai duomenis ieškomi pagal linijos pasirinktą kampą, pagal tai parenkama varža. Rutuliukas – kai ieškoma visų skaičiavimų mažiausia varža.

Pagal gautus rezultatus sunku įvertinti kokio jie tikslumo, kokią paklaidą daro šie skaičiavimai, bei kaip greitai jie skaičiuoja algoritmus. Todėl mano sukurtą platformą įvertinsime modeliuodami Simulink programa ir atliekant bandymus su skirtingais gedimo vietos nustatymo atstumo skaičiavimais. Bei naudojant sekančiame skyriuje (3.3 skyrius) aprašytą nuolatinės srovės dedamosios šalinimo filtrą ir be jo. Šie bandymai aprašyti 4 skyriuje.

### 3.3. Gedimo vietos nustatymas pašalinant nuolatinės srovės dedamąsias

Šiame skyriuje aprašytas gedimo vietos nustatymas panaudojant algoritmą, kuris pateiktas šio skyriaus apačioje, pašalinama nuolatinės srovės dedamoji (Algoritmas paremtas 2 – ame skyriuje aprašytomis metodikomis bei šaltiniais [2,10,6,7,8,12,10,13,14] ir kt.). Tai pagerina duomenų kokybę kadangi DC dedamosios reikšmės gedimo vietos nustatyme yra nereikalingos. Jos sukelia papildomus triukšmus, kurie iškraipo rezultatus. Tai sumažina gedimo vietos nustatymo algoritmų tikslumą.

Pašalinus DC dedamąją iš turimų duomenų, galima tiksliau apskaičiuoti gedimo vietą pagal 3.2 skyriuje aprašytą metodiką. Žemiau pateikiu grafikus bei lenteles, kurie akivaizdžiai parodo šio algoritmo svarbą. Bei puikiai iliustruoja kaip yra pašalinami triukšmai iš duomenų. Tai suteikia didesnes galimybes nustatyti tikslesnę gedimo vietą. Pilnas „MATLAB“ kodas pateiktas prieduose. Žr. priedas 3.

Toliau pritaikius skaičiavimus naudojamus šiame kode, gaunu išfiltruotus realius duomenis be DC dedamosios. Šiuos išfiltruotus duomenis pritaikau su 3.2 skyriuje naudojamu kodu ir metodika.

Rezultatai su realiais duomenimis (Alytus 1 matavimas), keičiant gedimo vietos apskaičiavimo metodiką, gaunami tokie:

1.) Kai gedimo vieta skaičiuojama pagal formules aprašytas 4.1.1 skyriuje (4.7 formulė gedimo vietai rasti) ir ieškomos mažiausios varžų reikšmės C fazės gedimo metu. Jos išrenkamos iš atvaizduotų vidutinių aktyviosios ir pilnutinės varžų reikšmių kitimo kreivių. Čia atsakymas lygus atstumui iki gedimo vietos 1:1.

$$Z_{sk} = 20,75, \Omega.$$

$$RR_{sk} = 37,2864, \Omega.$$

$$ZZ_{sk} = 75,4845, \Omega.$$

$$kint = \frac{ZZ_{sk} - RR_{sk}}{Z_{sk}} = \frac{75,1695 - 37,2864}{20,75} = 1,8409, Km.$$

čia,

$kint$  – atstumas iki gedimo vietos, km.

$ZZ_{sk}$  – apskaičiuota algoritmo pilnutinė varža, mažiausia gedimo metu,  $\Omega$ .

$RR_{sk}$  – apskaičiuota algoritmo aktyvioji varža, mažiausia gedimo metu,  $\Omega$ .

$Z_{sk}$  – linijos pilnutinė varža,  $\Omega$ .

2.) Kai gedimo vieta skaičiuojama pagal tokią formulę (4.11), ieškoma mažiausia pilnutinės varžos reikšmė gedimo metu C fazės. Kurios išrenkamos iš atvaizduotų vidutinių pilnutinės varžų reikšmių kitimo kreivių. Čia atsakymas lygus atstumui iki gedimo vietos 1:1.

$$Z_{sk} = 20,75, \Omega.$$

$$ZZ_{sk} = 75,4845, \Omega.$$

$$kint = \frac{ZZ_{sk}}{Z_{sk}} = \frac{75,4845}{20,75} = 3,6378, Km.$$

čia,

$kint$  – atstumas iki gedimo vietos, km.

3.) Kai skaičiuojama pagal tokią formulę (4.7 formulė gedimo vietai rasti), yra ieškoma aktyvioji ir pilnutinė varžos pagal linijos kampą kuris yra 64 laipsniai. Tada išrenkamos mažiausios reikšmės tuo pačiu momentu, kai yra toks linijos kampas. Čia atsakymas lygus atstumui iki gedimo vietos 1:1.

Rezultatai gaunami tokie:

$$Z_{sk} = 20,75, \Omega.$$

$$RR_{sk} = 40,8074, \Omega.$$

$$ZZ_{sk} = 93,0887, \Omega.$$

$$kint = \frac{ZZ_{sk} - RR_{sk}}{Z_{sk}} = \frac{93,0887 - 40,8074}{20,75} = 2,5196, Km.$$

čia,

*kint* – atstumas iki gedimo vietos, km.

4.) Kai gedimo vieta skaičiuojama pagal tokią formulę (4.11) pilnutinė varža imama pagal linijos kampą kuris yra 64 laipsniai. Tada ieškoma mažiausios jos reikšmės, pagal parinktą linijos kampą.

Rezultatai gaunami tokie:

$$Z_{sk} = 20,75, \Omega.$$

$$ZZ_{sk} = 93,0887, \Omega.$$

$$kint = \frac{ZZ_{sk}}{Z_{sk}} = \frac{93,0887}{20,75} = 4,4862, Km.$$

čia,

*kint* – atstumas iki gedimo vietos, km.

Taip pat perskaičiuojami laikai gedimo metu jie gaunami tokie:

Tavarijos = 0.0434 0.9793, čia avarijos laikas, s.

dTavarijos = 0.9359, avarijos trukmė, s.

Taki = 0.9806 2.9042, čia AKĮ pauzės laikas, s.

dTaki = 1.9236, čia AKĮ trukmė, s.

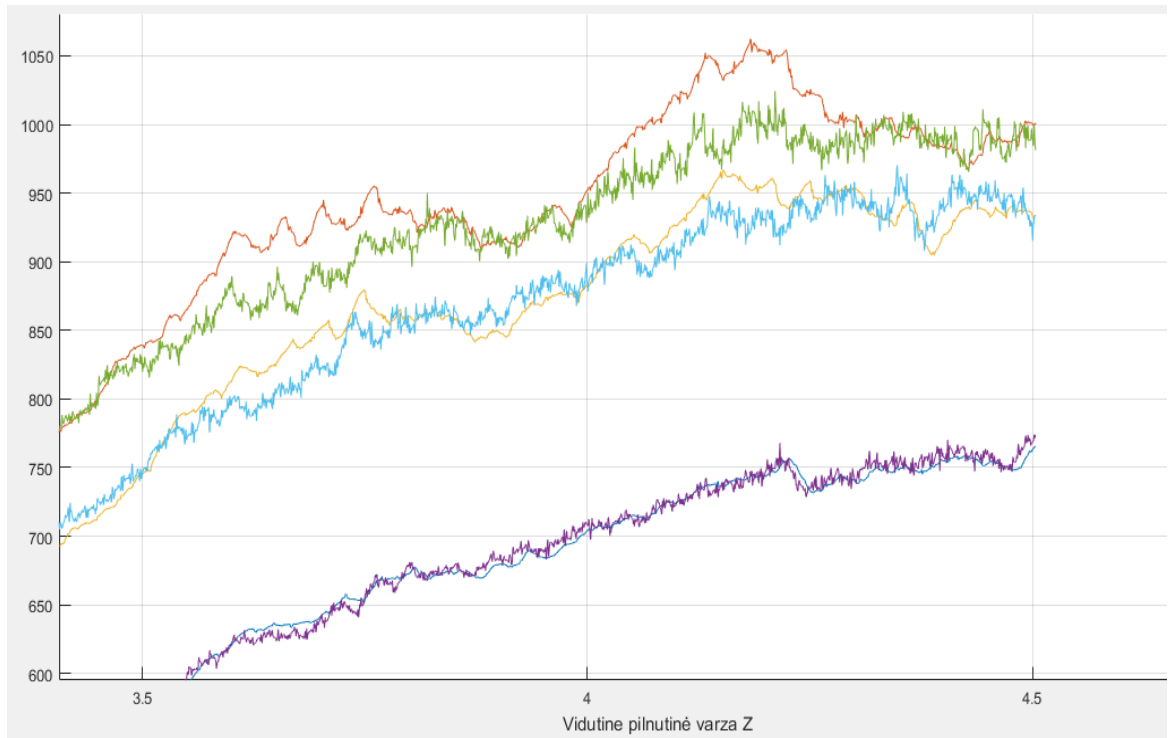
Palyginimui kaip DC dedamosios pašalinimas įtakoja skaičiuojamus duomenis ir šalina triukšmus pateikiu dar keletą grafikų.

Šiuose grafikuose žymėjimas toks: A fazė – violetinė spalva. B fazė – žalia spalva. C fazė – žydra spalva (3.3.1 – 3.3.4 pav.). Kai duomenys atvaizduojami be DC filtro su neišfiltruotais duomenimis.

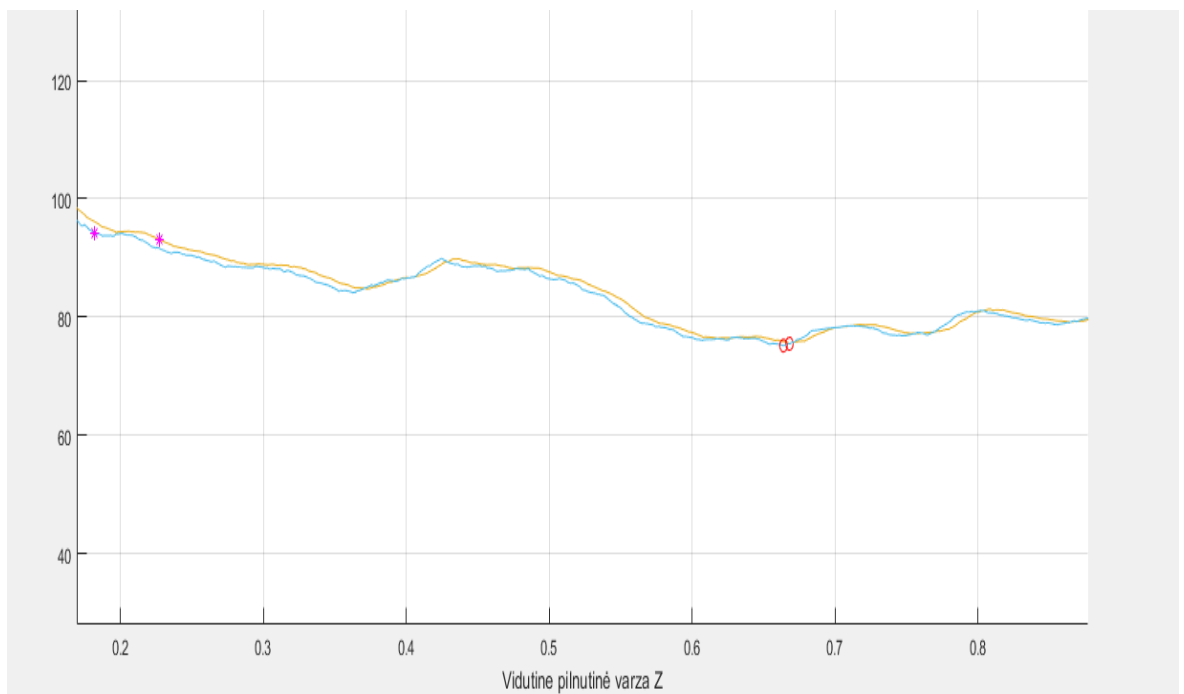
Bei, kai duomenys išfiltruoti ir pritaikytas nuolatinės srovės dedamosios šalinimas iš duomenų. Grafikuose žymėjimas toks: A fazė – melyna spalva. B fazė – oranžinė spalva. C fazė – geltona spalva



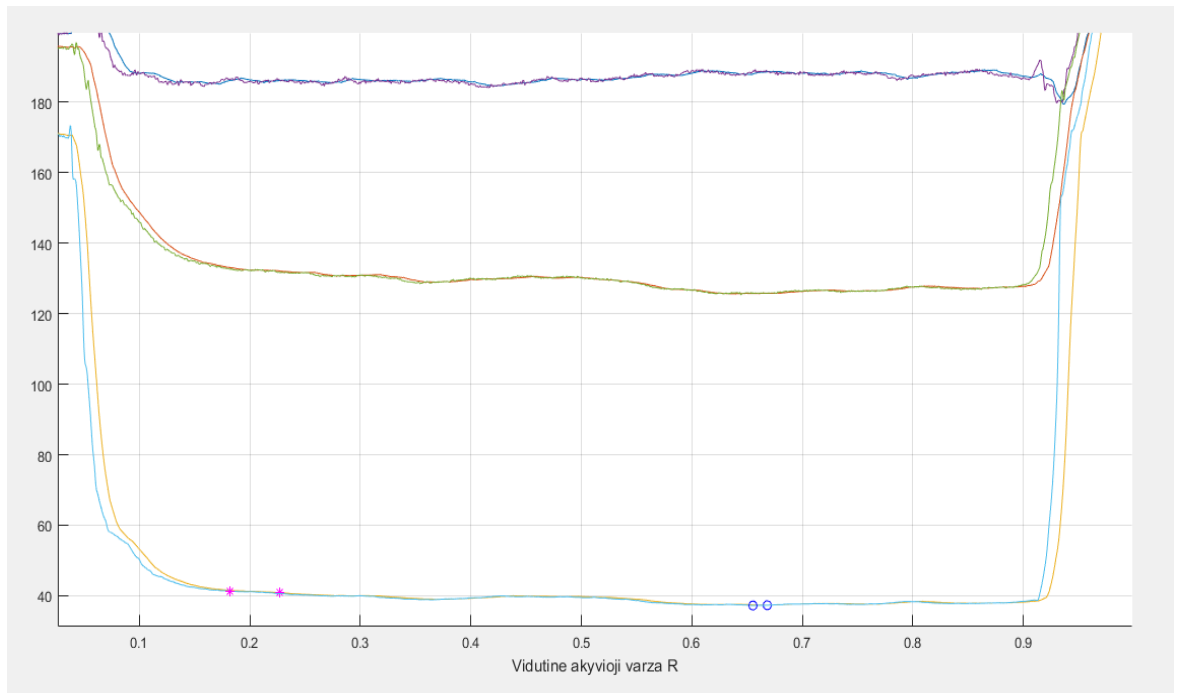
(3.3.1 – 3.3.4 pav.). Žvaigždutė žymi kai duomenis ieškomi pagal linijos pasirinktą kampą, pagal tai parenkama varža. Rutuliukas – kai ieškoma visų skaičiavimų mažiausia varža.



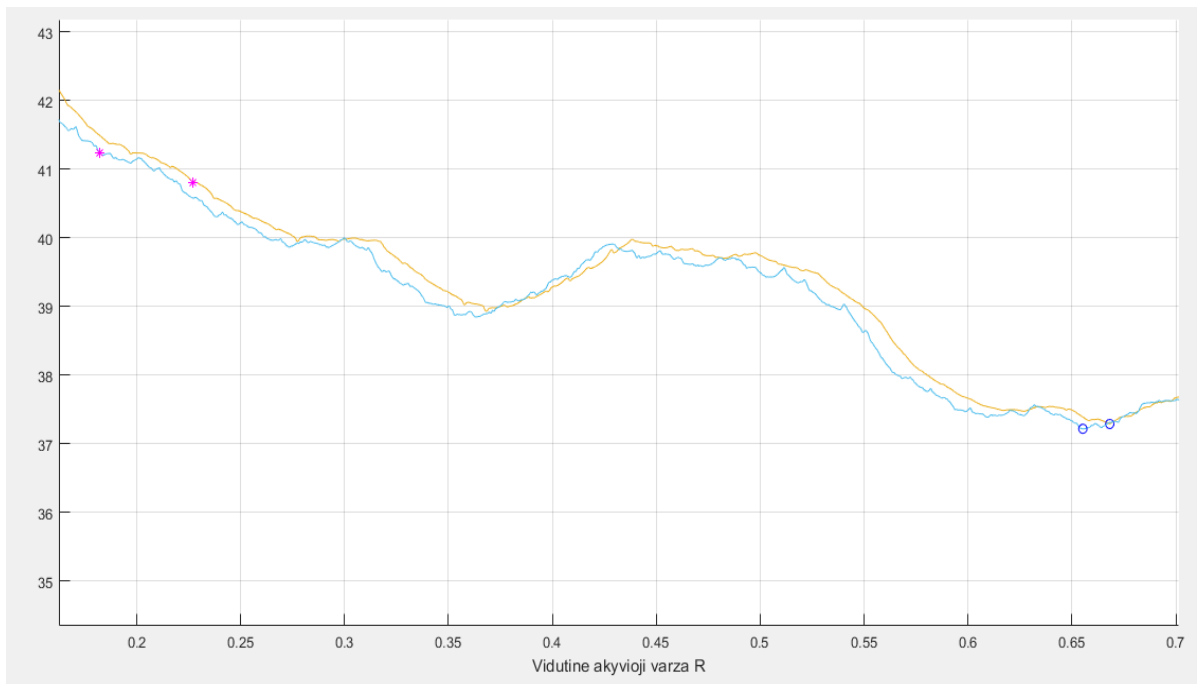
3.3.1 pav. Vidutinė pilnutinė varža kai duomenys išfiltruoti su DC filtru ir be jų. Filto įtaka duomenims (Alytus 1 matavimas).



3.3.2 pav. Vidutinė pilnutinė varža kai duomenys išfiltruoti su DC filtru ir be jų. Filto įtaka varžos parinkimui (Alytus 1 matavimas).



3.3.3 pav. Vidutinė aktyvioji varža kai duomenys išfiltruoti su DC filtru ir be jų. Filto įtaka duomenims (Alytus 1 matavimas).



3.3.4 pav. Vidutinė aktyvioji varža kai duomenys išfiltruoti su DC filtru ir be jų. Filto įtaka varžos parinkimui (Alytus 1 matavimas).

Iš viršuje pateiktų grafikų (3.3.1 – 3.3.4 pav.) puikiai matome nuolatinės srovės dedamosios šalinimo filtro įtaką duomenims ir varžų parinkimui. Matome kaip pašalinami triukšmai iš naudojamų duomenų skaičiavimuose. Bei kaip kinta varžos parinkimas naudojant duomenis be filtro ir su juo.

Tam, kad galėtume įvertinti ar tikrai šis filtras pagerina modelio tikslumą ar jį pablogina, bei kaip greitai galima atlikti skaičiavimus su DC filtru, atliksime bandymus su sumodeliuota sistema Simulink programa. Imituosime gedimus bei keisime atstumus iki gedimo vietos. Bandymų metu nustatysime kokius tikslus yra mano pateikiami skaičiavimai ir metodai, kokius jie greiti, bei kokią paklaidą jie daro skaičiavimuose. Taip pat pamatysime kokią įtaką daro atstumas iki gedimo vietos šių skaičiavimų tikslumui. Šie bandymai aprašyti 4 skyriuje. Atlikus šiuos bandymus galėsiu palyginti kaip efektyviai galima pritaikyti mano pateikiamus metodus su realiais duomenimis esant tikriems gedimams elektros energetikos sistemoje.

#### 4. SIMULINK MODELIAVIMAS

Šiame skyriuje aprašyta kaip buvo tiriamas gedimo vietos nustatymo algoritmo tikslumas. Simulink programos pagalba buvo sumodeliuota aukštos įtampos elektros energetikos sistema, bei įmituotas gedimas su Simulink programos pagalba. Gedimai įmituoti fazė – fazė, fazė – žemė, dvifaziai gedimai. Simulink programos pagalba duomenys apie gedimą buvo perkelti ir pritaikyti su jau aukščiau aprašytais algoritmais. Bei žinant tikslią gedimo vietą nustatytas jų tikslumas, bei skaičiavimų paklaidos. Nustatytas metodikų skaičiavimų greitis. Sumodeliuota sistema, bei gauti duomenys pateikti žemiau. Šiuose bandymuose vertinu tik linijos ir gedimo varžas t.j.metu. Šias varžas nusistatau rankiniu būdu Simulink programoje ir apsiskaičiuoju jas taip ( naudojami šaltiniai [2,6,7,20,1,3,4,8,19] ir kt. ) pagal „MATLAB“ pateikiamą metodiką:

$$R_F = atstumas \cdot r, \Omega \quad (4.1);$$

$$X_0 = atstumas \cdot x_L, H \quad (4.2);$$

$$R_0 = atstumas \cdot r_0, \Omega \quad (4.3);$$

$$R = R_F + R_0, \Omega \quad (4.4);$$

$$X_L = 2\pi f \cdot X_0, \Omega \quad (4.5);$$

$$Z = \sqrt{R^2 + X_L^2}, \Omega \quad (4.6);$$

čia,

$$x_L = 0,9337 \cdot 10^{-3}, H.$$

$$r_0 = 0,0127, \Omega/Km.$$

$$r = 0,1, \Omega/Km.$$

$$f = 50, Hz.$$

*atstumas* – tai atstumas iki gedimo vietos, Km.

*r* – gedimo varža,  $\Omega/km$ .

*f* – sistemos dažnis, Hz.

*R* – linijos aktyvioji varža,  $\Omega$ .

*X<sub>L</sub>* – linijos reaktyvioji varža,  $\Omega$ .

*Z* – linijos pilnutinė varža,  $\Omega$ .

Toliau apskaičiuoju varžas. Jei bandyme nevertiname gedimo varžos, tai gedimo metu visą aktyviąją varžą sudaro tik aktyvioji linijos varža.

Matavimo 3 skaičiavimai kai atstumas iki gedimo 130 km (Bandymas Nr.1).

čia,

$$R_F = 1,3 \cdot 0,1 = 0,13, \Omega$$

$$X_0 = 1,3 \cdot 0,9337 \cdot 10^{-3} = 0,0012138 \text{ H}$$

$$R_0 = 1,3 \cdot 0,0127 = 0,01651, \Omega$$

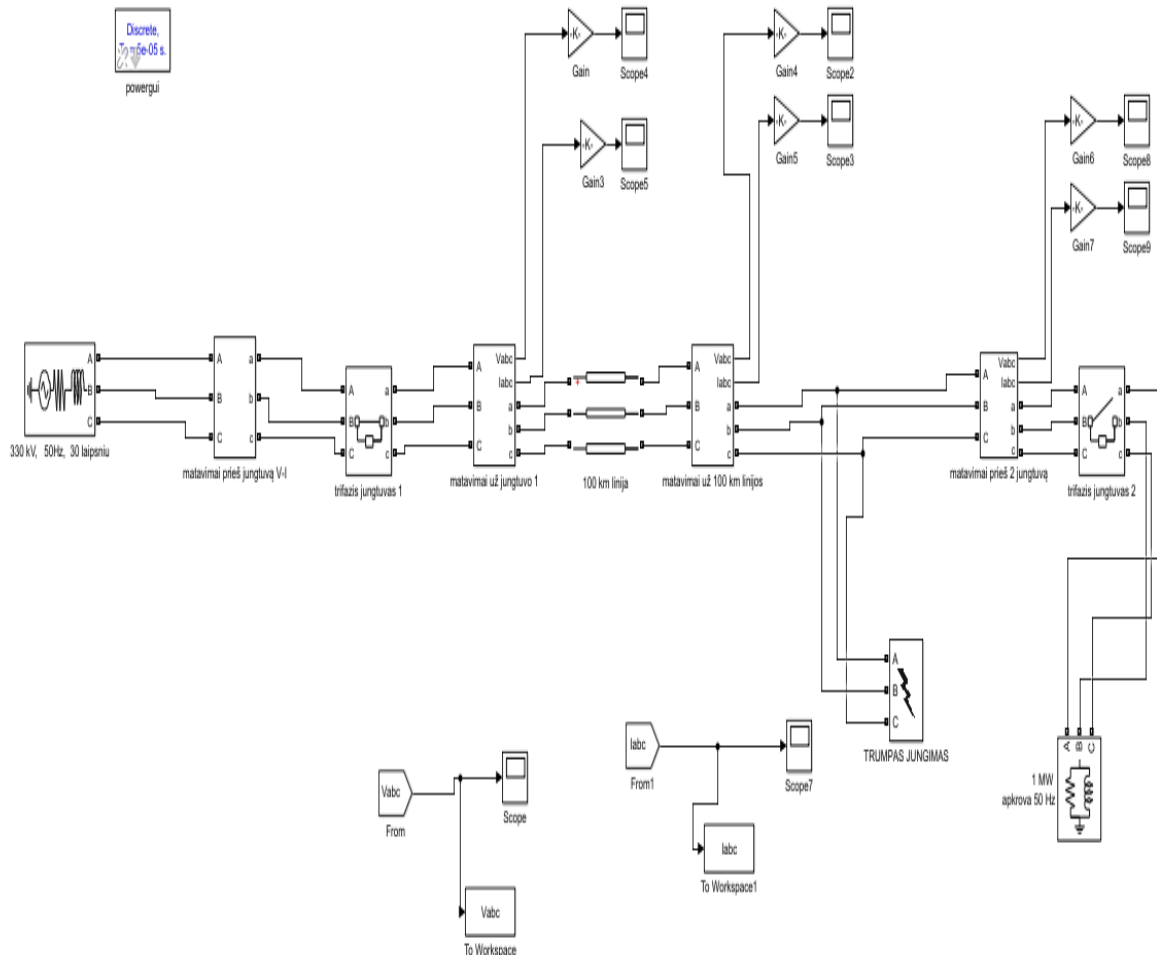
$$R = 0,13 + 0,01651 = 0,14651, \Omega$$

$$X_L = 2 \cdot 3,14 \cdot 50 \cdot 0,0012138 = 0,38113, \Omega$$

$$Z = \sqrt{R^2 + X_L^2} = \sqrt{0,14651^2 + 0,38113^2} = 0,4083, \Omega$$

Visi kiti skaičiavimai bandymuose atliekami analogiškai ieškant pilnutinės linijos varžos.

Simulink programa modeliuojama tokia schema:



4.1 pav. Principinė modeliuojamo gedimo schema Simulink programoje.

Čia modeliuojama 330 kV linija kai atstumas iki gedimo vietos yra nuo 70 – 190 Km. Apkrova 1 MW. Sistemos dažnis 50 Hz, naudojami 2 jungtuvai linijos pradžioje ir linijos pabaigoje. Čia

modeliuojami gedimas fazė – fazė, fazė – žemė, fazė – fazė – žemė, 3 fazės ir 3 fazės – žemė. Matavimų duomenys perkeliama į „MATLAB“ programą kurioje srovių ir įtampų duomenys gedimo metu persikelia į vieną duomenų failą. Pagal laiko momentu kintačius pereinamuosius procesų duomenis nustatinėjau gedimo vietą. Pritaikant prieduose pateiktus „MATLAB“ kodus (metodikas). Žr. Priedas 1,2,3.

#### **4.1. Gedimo vietos algoritmų tikslumo, paklaidos ir greičio skaičiavimų bandymai**

##### **4.1.1. Bandymas Nr.1 gedimas fazė – žemė be DC filtro**

Šiame skyriuje aprašytas tyrimas kai gedimo vietos nustatymo algoritmas naudojamas be nuolatinės srovės dedamosios pašalinimo filtro (DC filtro). Algoritmas nustato gedimo vietos atstumą pagal žemiau pateiktą formulę. Bandymą atliksiu 5 kartus su skirtingomis atstumų reikšmėmis, kai gedimo vieta yra 70,100,130,160,190 km, atstumuose. Bandymo metu nustatysiu koks yra šio algoritmo tikslumas keičiant atstumus iki gedimo vietos. Taip pat, kaip greitai skaičiuojamas algoritmas (kiek laiko užtrunka skaičiavimai). Bei nustatysiu šio algoritmo paklaidą, kiek procentų jis suklysta nuo tikrosios gedimo vietos. Gedimas vyksta tarp fazės C ir žemės.

Gedimo vieta skaičiuojama pagal 3.2 naudojamą metodiką. Kai gedimo vietos atstumas surandamas pagal formulę:

$$kint = \frac{ZZ_{sk} - RR_{sk}}{Z_{sk}} \quad (4.7);$$

čia,

$ZZ_{sk}$  – apskaičiuota algoritmo pilnutinė varža mažiausia gedimo metu,  $\Omega$ .

$RR_{sk}$  – apskaičiuota algoritmo aktyvioji varža mažiausia gedimo metu,  $\Omega$ .

$Z_{sk}$  – linijos pilnutinė varža,  $\Omega$ .

Kai gedimo vieta pasirinkta 70 km. Tada atstumas iki gedimo vietos pagal šį tiriamą algoritmą būtų surastas taip:

$$kint = \frac{ZZ_{sk} - RR_{sk}}{Z_{sk}} = \frac{37,344 - 21,945}{21,98} = 0,70591$$

tada apskaičiuotas atstumas iki gedimos vietos:

$$kint \cdot 100 = 0,70591 \cdot 100 = 70,591 \text{ Km}$$

Rezultatuose  $kint = 0,01$  reiškia 1 km, todėl 4.1 lentelėje pateikti rezultatai yra padauginti iš 100, kad lengviau suprasti jų tikrąją vertę.

Gedimo vietos nustatymo algoritmo tikslumas surandamas taip:

$$tikslumas_1 = \frac{apsk.atstumas \cdot 100}{tikrasis\ gedimo\ vietos\ atstumas} = \frac{70,591 \cdot 100}{70} = 100,884, \% \quad (4.8);$$

$$tikslumas\ tikrasis = 200 - tikslumas_1 = 200 - 100,884 = 99,155 \% \quad (4.9);$$

Tada paklaida bus surandama taip:

$$paklaida = 100 - tikslumas\ tikrasis = 100 - 99,155 = 0,845, \% \quad (4.10);$$

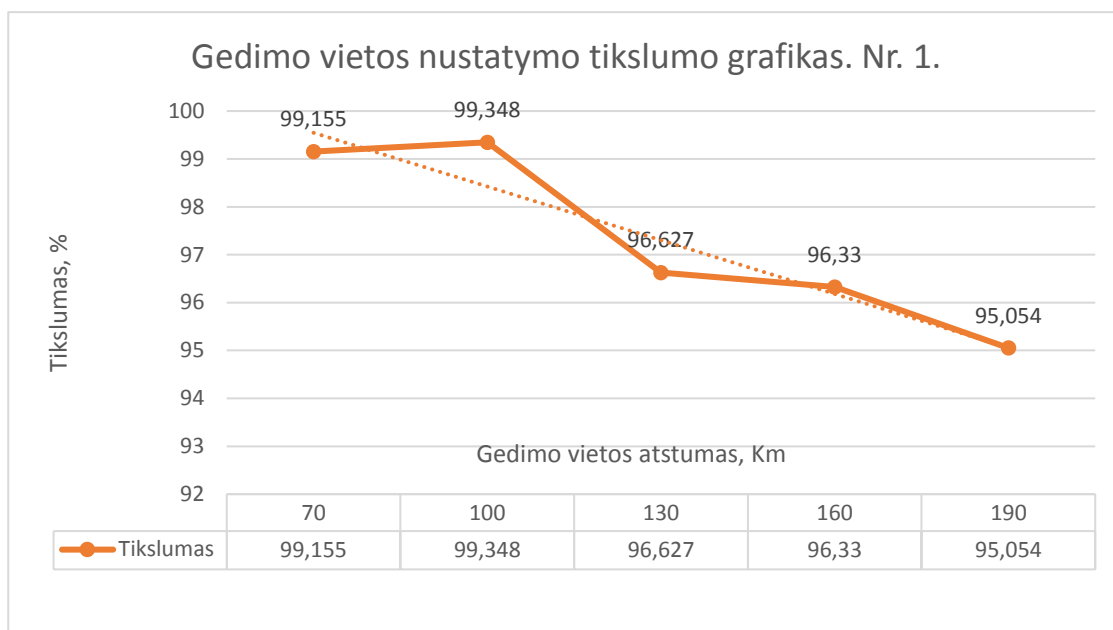
Algoritmo skaičiavimo greitis (skaičiavimo laikas) surandamas rankiniu būdu chronometro pagalba. Toliau pateikiama lentelė, kur surašyta visų matavimų duomenys, bei rezultatai gauti skirtingų matavimų metu. Taip pat apskaičiuoti algoritmo tikslumo, skaičiavimo greičio ir algoritmo skaičiavimo paklaidos vidurkiai.

4.1. lentelė. Bandyto Nr. 1 rezultatų lentelė.

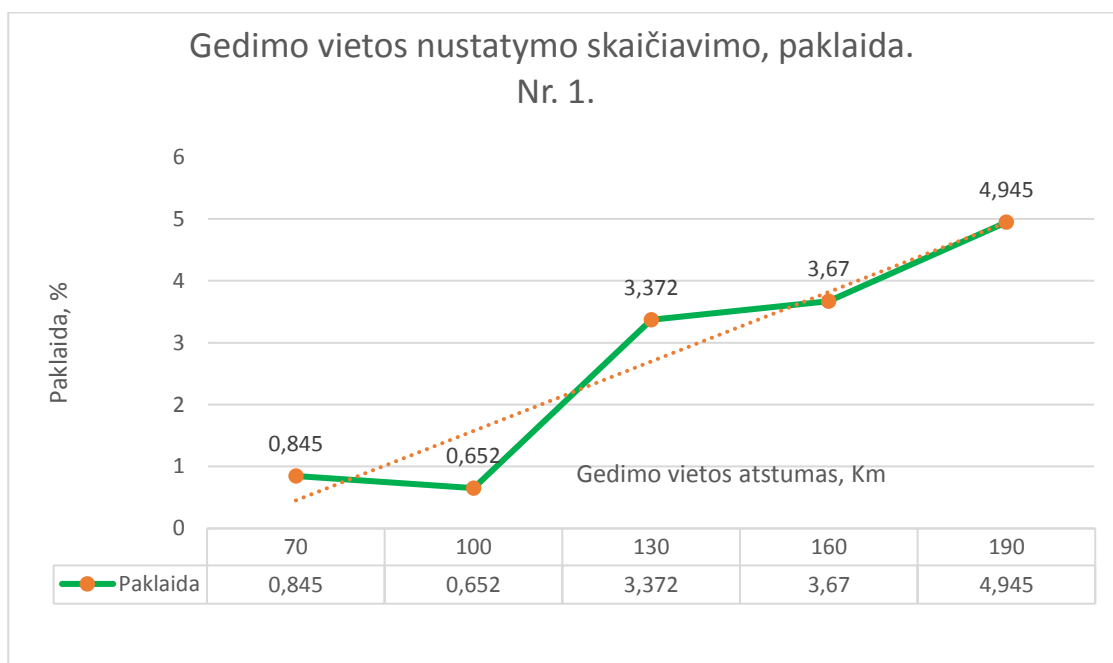
<i>Matavimo Nr.</i>	<i>Zsk, Ω</i>	<i>RRsk, Ω</i>	<i>ZZsk, Ω</i>	<i>Skaičiavimo laikas, s.</i>	<i>Atstumas iki gedimo, km.</i>	<i>Tikslumas, %.</i>	<i>Paklaida, %.</i>	<i>Tikras atstumas iki gedimo, km.</i>
Matavimas 1	21,98	21,945	37,344	20,14	70,591	99,155	0,845	70
Matavimas 2	31,40	38,994	70,599	20,61	100,652	99,348	0,652	100
Matavimas 3	40,83	44,152	99,021	21,93	134,384	96,627	3,372	130
Matavimas 4	48,21	48,553	128,52	22,43	165,872	96,330	3,670	160
Matavimas 5	59,67	59,22	178,20	24,88	199,396	95,054	4,945	190
<i>Vidutinis skaičiavimo greitis</i>	x	x	x	<b>21,998</b>	x	x	x	x
<i>Viso vidutinis tikslumas, paklaida</i>	x	x	x	x	x	<b>97,303</b>	<b>2,696</b>	x

Toliau pateikiu grafikus, kurie sudaryti iš viršuje esančios 1 lentelės duomenų bei rezultatų.

Pateikiu gedimo vietos nustatymo algoritmo tikslumo bei skaičiavimo paklaidos grafikus keičiantis gedimo vietos atstumui.



4.1.1 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos.



4.1.2 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos.



Iš šių grafikų puikiai matome, kad gedimo vietos nustatymo algoritmo tikslumas mažėja didėjant atstumui iki gedimo vietos. Punktyrinė linija parodo duomenų tendenciją. Šio algoritmo vidutinis tikslumas, keičiantis atstumui iki gedimo vietos nuo 70 iki 190 km., yra 97,303 %, o šio algoritmo skaičiavimo paklaida yra 2,696 %. Mano tiriamajame bandyme gedimo varža yra žinoma. Ji sudaro 6,551 % visos linijos varžos gedimo metu (pagal 4.1-4.6 formules). Ją nustatome rankiniu būdu „Simulink“ programoje.

Taip pat nustatytas vidutinis skaičiavimo greitis šio bandymo metu yra 21,998 sekundės.

Šio algoritmo tikslumą, kai žinoma gedimo varža, vertinu gerai, nes tikslumas viršija 97 %.

#### **4.1.2. Bandymas Nr.2 gedimas fazė – žemė su DC filtru**

Šiame skyriuje aprašytas tyrimas kai gedimo vietos nustatymo algoritmas naudojamas kartu su nuolatinės srovės dedamosios pašalinimo filtru (DC filtru). Ir algoritmas apskaičiuoja gedimo vietos atstumą pagal 4.1.1. skyriuje pateiktą (4.7) formulę. Bandymą atliksiu 5 kartus su skirtingomis atstumų reikšmėmis, kai gedimo vieta yra 70,100,130,160,190 km, atstumuose. Bandymo metu nustatysiu koks yra šio algoritmo tikslumas keičiant atstumus iki gedimo vietos. Taip pat, kaip greitai skaičiuojamas algoritmas (kiek laiko užtrunka skaičiavimai). Bei apskaičiuosiu šio algoritmo paklaidą, kiek procentų jis suklysta nuo tikrosios gedimo vietos. Gedimas vyksta tarp fazės C ir žemės.

Gedimo vieta skaičiuojama pagal 3.2 ir 3.3 naudojamas metodikas. Gedimo vietos atstumas surandamas pagal 3.2 skyriuje aprašytą algoritimą, kuris skaičiuoja mažiausią pilnutinę ir aktyviają varžą, kurių reikšmės išrenka iš apskaičiuotų varžų vidurkių kreivių. Tada paskaičiuotiems duomenims pritaikomas 3.3 filtras ir perskaičiuojamos vidutinės varžų reikšmės. Pagal išfiltruotus duomenis pritaikomas 3.2 sk. aprašytas algoritmas, kuris iš vidutinių varžų kreivių suranda mažiausias vertes, bei perskaičiuojama gedimo vieta su jau išfiltruotais duomenimis pagal formulę (4.7).

Gedimo vietos nustatymo paklaida ir tikslumas procentaliai apskaičiuojami pagal tas pačias metodikas, kaip ir bandyme Nr. 1 (4.1.1 skyriuje, formulės 4.8, 4.9 ir 4.10).

Algoritmo skaičiavimo greitis (skaičiavimo laikas) surandamas rankiniu būdu chronometro pagalba.

Toliau pateikiama lentelė, kur surašyta visų matavimų duomenys bei rezultatai gauti skirtingų matavimų metu.

Taip pat, apskaičiuoti algoritmo tikslumo, skaičiavimo greičio ir algoritmo skaičiavimo paklaidos vidurkiai.

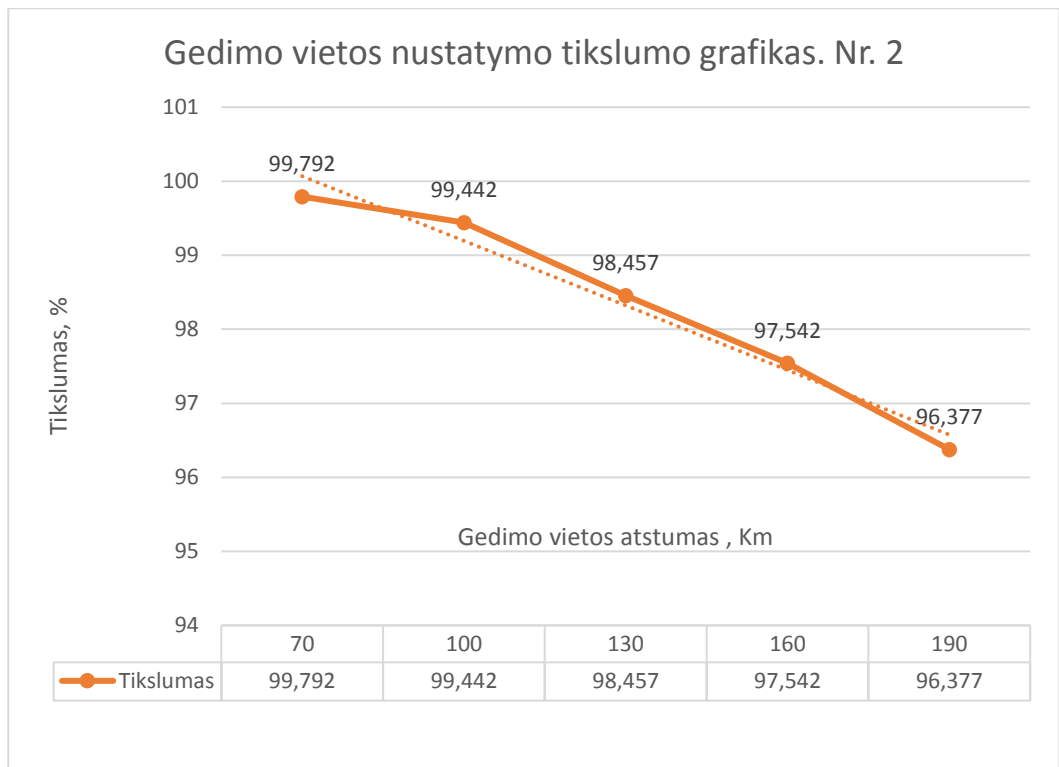
Po lentele toliau atvaizduoju grafikus, kurie sudaryti iš virš grafikų esančios lentelės duomenų bei rezultatų. Žr.2 lentelė.

Pateikiu gedimo vietos nustatymo algoritmo tikslumo bei skaičiavimo paklaidos grafikus keičiantis gedimo vietos atstumui.

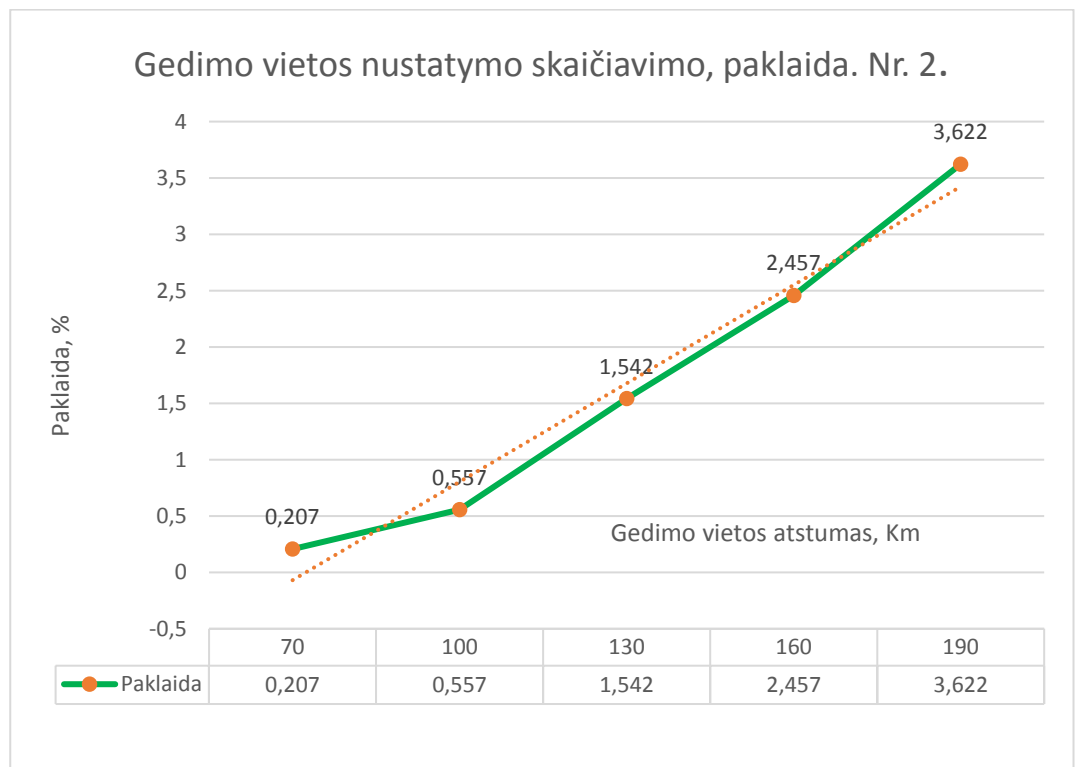
Čia Nr. 2 bandymo rezultatų lentelė.

4.2.lentelė. Bandymo Nr.2 rezultatų lentelė.

<i>Matavimo Nr.</i>	<i>Zsk, Ω</i>	<i>RRsk, Ω</i>	<i>ZZsk, Ω</i>	<i>Skaičiavimo laikas, s.</i>	<i>Atstumas iki gedimo, km.</i>	<i>Tikslumas %.</i>	<i>Paklaida %.</i>	<i>Tikrasis atstumas iki gedimo, km.</i>
Matavimas 1	21,98	21,935	37,289	80,84	69,854	99,792	0,207	70
Matavimas 2	31,40	38,924	70,499	81,61	100,557	99,442	0,557	100
Matavimas 3	40,83	44,152	99,021	79,85	132,005	98,457	1,542	130
Matavimas 4	48,21	48,201	127,233	80,12	163,932	97,542	2,457	160
Matavimas 5	59,67	58,82	176,30	82,18	196,882	96,377	3,622	190
<i>Vidutinis skaičiavimo greitis</i>	x	x	x	<b>80,92</b>	x	x	x	x
<i>Viso vidutinis tikslumas, paklaida</i>	x	x	x	x	x	<b>98,322</b>	<b>1,678</b>	



4.1.3 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras.



4.1.4 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras.

Iš šių grafikų puikiai matome, kad gedimo vietos nustatymo algoritmo tikslumas mažėja didėjant atstumui iki gedimo vietos. Punktyrinė linija parodo duomenų tendenciją. Šio algoritmo vidutinis tikslumas, keičiantis atstumui iki gedimo vietos nuo 70 iki 190 km., yra 98,322 %, o šio algoritmo skaičiavimo paklaida yra 1,678 %. Šiame bandyme gedimo varža yra žinoma. Ji sudaro 6,551 % visos linijos varžos gedimo metu (pagal 4.1-4.6 formules). Ją nustatome rankiniu būdu „Simulink“ programoje.

Taip pat nustatytas vidutinis skaičiavimo greitis šio bandymo metu yra 80,92 sekundės.

Taigi iš šio bandymo matome, kad skaičiavimo modelis, kur naudojamas nuolatinės srovės dedamosios šalinimo filtras, yra lėtesnis viena minute nei pirmojo bandymo metu, kur filtras nenaudojamas. Tačiau šio modelio skaičiavimo paklaida mažesnė vienu procentu, taigi pagerinamas gedimo vietos nustatymo algoritmo tikslumas vienu procentu. Taip pasiekiamo aukštą tikslumą. Daugiau nei 98 %.

#### ***4.1.3. Bandymas Nr.3 gedimas fazė – žemė su DC filtru kai nežinoma gedimo varža***

Šiame skyriuje aprašytas tyrimas, kai gedimo vietos nustatymo algoritmas naudojamas kartu su nuolatinės srovės dedamosios pašalinimo filtru (DC filtru), tačiau nėra žinoma gedimo varža. Rankiniu būdu nustačiau „Simulink“ programoje, kad ji sudarytu 6,551 % visos linijos varžos.

Pagal šį bandymą galėsime vertinti modelio tikslumą, jei nežinome linijos varžos. Taip pat, pagal šių rezultatų tikslumą galėsime atsižvelgti kokie šio modelio skaičiavimai yra su realiais duomenimis ir kiek skaičiavimai yra tikslūs.

Algoritmas nustato gedimo vietos atstumą pagal 4.1.1. skyriuje pateiktą (4.7) formulę. Bandymą atliksiu 5 kartus su skirtingomis atstumų reikšmėmis, kai gedimo vieta yra 70,100,130,160,190 km, atstumuose. Bandymo metu nustatysiu koks yra šio algoritmo tikslumas keičiant atstumus iki gedimo vietos. Taip pat, kaip greitai skaičiuojamas algoritmas (kiek laiko užtrunka skaičiavimai). Bei nustatysiu šio algoritmo paklaidą, kiek procentų jis suklysta nuo tikrosios gedimo vietos. Gedimas vyksta tarp fazės C ir žemės.

Gedimo vieta skaičiuojama pagal tą pačią metodiką kaip 2 bandyme, tik skiriasi linijos pilnutinė varža. Algoritmo skaičiavimo greitis (skaičiavimo laikas) surandamas rankiniu būdu chronometro pagalba.

Toliau pateikiama lentelė, kur surašyta visų matavimų duomenys bei rezultatai gauti skirtingų matavimų metu.

Taip pat, apskaičiuoti algoritmo tikslumo, skaičiavimo greičio ir algoritmo skaičiavimo paklaidos vidurkiai.

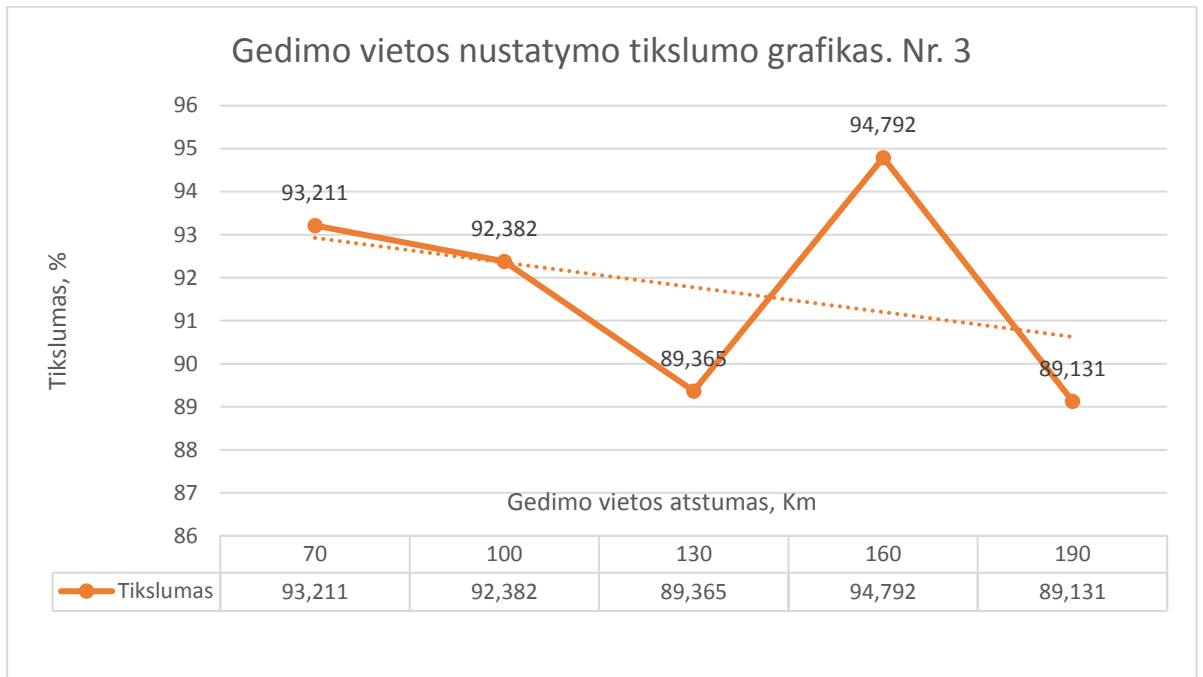
Po lentelė toliau atvaizduoja grafikus, kurie sudaryti iš virš grafikų esančios lentelės duomenų bei rezultatų. Žr. 4.3 lentelė.

Čia Nr. 3 bandymo rezultatų lentelė.

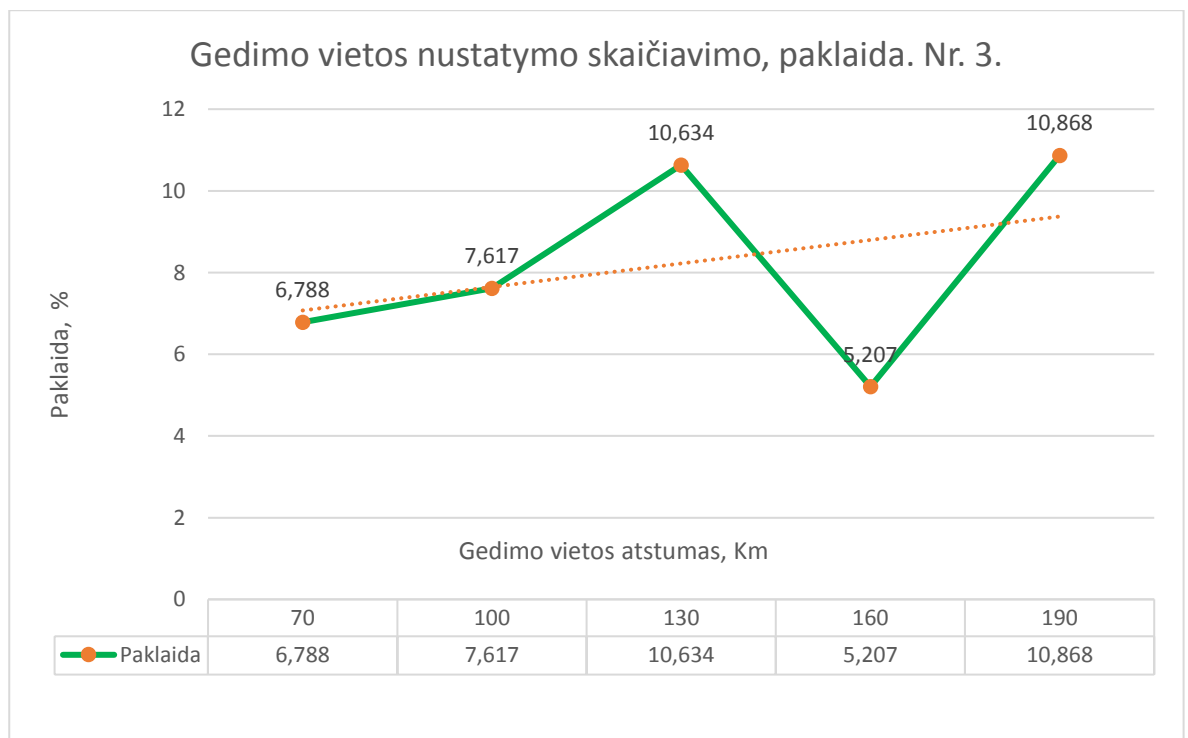
4.3.lentelė. Bandymo Nr.3 rezultatų lentelė.

<i>Matavimo Nr.</i>	<i>Zsk, Ω</i>	<i>RRsk, Ω</i>	<i>ZZsk, Ω</i>	<i>Skaičiavimo laikas, s.</i>	<i>Atstumas iki gedimo km.</i>	<i>Tikslumas %.</i>	<i>Paklaida %.</i>	<i>Tikrasis atstumas iki gedimo, km.</i>
Matavimas 1	20,54	21,935	37,289	80,12	74,751	93,211	6,788	70
Matavimas 2	29,34	38,924	70,499	81,98	107,617	92,382	7,617	100
Matavimas 3	38,15	44,152	99,021	80,01	143,824	89,365	10,634	130
Matavimas 4	46,95	48,201	127,233	79,18	168,332	94,792	5,207	160
Matavimas 5	55,77	58,82	176,30	82,55	210,650	89,131	10,868	190
<i>Vidutinis Skaičiavimo greitis</i>	x	x	x	<b>80,76</b>	x	x	x	x
<i>Viso vidutinis tikslumas, paklaida</i>	x	x	x	x	x	<b>91,777</b>	<b>8,222</b>	x

Pateikiu gedimo vietos nustatymo algoritmo tikslumo bei skaičiavimo paklaidos grafikus keičiantis gedimo vietos atstumui. Kai nėra vertinama gedimo varža.



4.1.5 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras, bet nevertinama gedimo varža.



4.1.6 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras, bet nevertinama gedimo varža.

Iš šių grafikų puikiai matome, kad gedimo vietos nustatymo algoritmo tikslumas mažėja didėjant atstumui iki gedimo vietos. Punktyrinė linija parodo duomenų tendenciją. Šio algoritmo vidutinis tikslumas, keičiantis atstumui iki gedimo vietos nuo 70 iki 190 km., yra 91,777 %, o šio algoritmo skaičiavimo paklaida yra 8,222 %. Šiame bandyme gedimo varža skaičiavime nevertinta. Ji sudaro 6,551 % visos linijos varžos gedimo metu, todėl viso algoritmo tikslumas gerokai sumažėja. Gedimo varžą nustatome rankiniu būdu „Simulink“ programoje.

Taip pat nustatytas vidutinis skaičiavimo greitis šio bandymo metu yra 80,76 sekundės. Taigi skaičiavimo laikas yra praktiškai toks pat kaip ir 2 – ojo bandymo ir minute didesnis nei pirmajame bandyme.

Taigi iš šio bandymo matome, kad šis skaičiavimo modelis, kur naudojamas nuolatinės srovės dedamosios šalinimo filtras bei nevertinama gedimo varža, yra mažesnio tikslumo. Ši paklaida priklauso nuo gedimo varžos dydžio, ji įtakoja visų duomenų pilnutinę linijos varžą. Dėl to skaičiavimas tampa mažiau tikslus. Skaičiavimo tikslumas šio bandymo metu viršija 90 %, tai galėtume vertinti gerai.

#### ***4.1.4. Bandymas Nr.4 gedimas fazė – žemė be DC filtro kai gedimo vietos atstumas skaičiuojamas pagal linijos kampą***

Šiame skyriuje aprašytas tyrimas kai gedimo vietos nustatymo algoritmas naudojamas be nuolatinės srovės dedamosios pašalinimo filtro (DC filtro). Ir algoritmas nustato gedimo vietos atstumą pagal formulę (4.7), tačiau gedimo vieta surandama ne pagal mažiausią gedimo pilnutinę ir aktyviąją varžą, o pagal linijos kampą. Ieškomos mažiausios pilnutinės ir aktyviosios varžos, kurios atitiktų pasirinktą linijos kampą. Taigi varžos nebūtinai gali būti mažiausios. Šiame bandyme gedimo varža yra žinoma. Bandymą atliksiu 5 kartus su skirtingomis atstumų reikšmėmis, kai gedimo vieta yra 70,100,130,160,190 km. atstumuose. Bandymo metu nustatysiu koks yra šio algoritmo tikslumas keičiant atstumus iki gedimo vietos. Taip pat, kaip greitai skaičiuojamas algoritmas (kiek laiko užtrunka skaičiavimai). Bei nustatysiu šio algoritmo paklaidą, kiek procentų jis suklysta nuo tikrosios gedimo vietos. Gedimas vyksta tarp fazės C ir žemės. Skaičiavimo algoritmo metodika pateikta 3.2 skyriuje. Matavimų linijos kampas yra 31,5 laipsnis.

Gedimo vietos nustatymo paklaida ir tikslumas procentaliai apskaičiuojami pagal tas pačias metodikas kaip ir bandyme Nr. 1 (4.1.1 skyriuje, formulės 4.8, 4.9 ir 4.10).

Algoritmo skaičiavimo greitis (skaičiavimo laikas) surandamas rankiniu būdu chronometro pagalba.

Toliau pateikiama lentelė, kur surašyti visų matavimų duomenys, bei rezultatai gauti skirtingų matavimų metu.

Taip pat, apskaičiuoti algoritmo tikslumo, skaičiavimo greičio ir algoritmo skaičiavimo paklaidos vidurkiai.

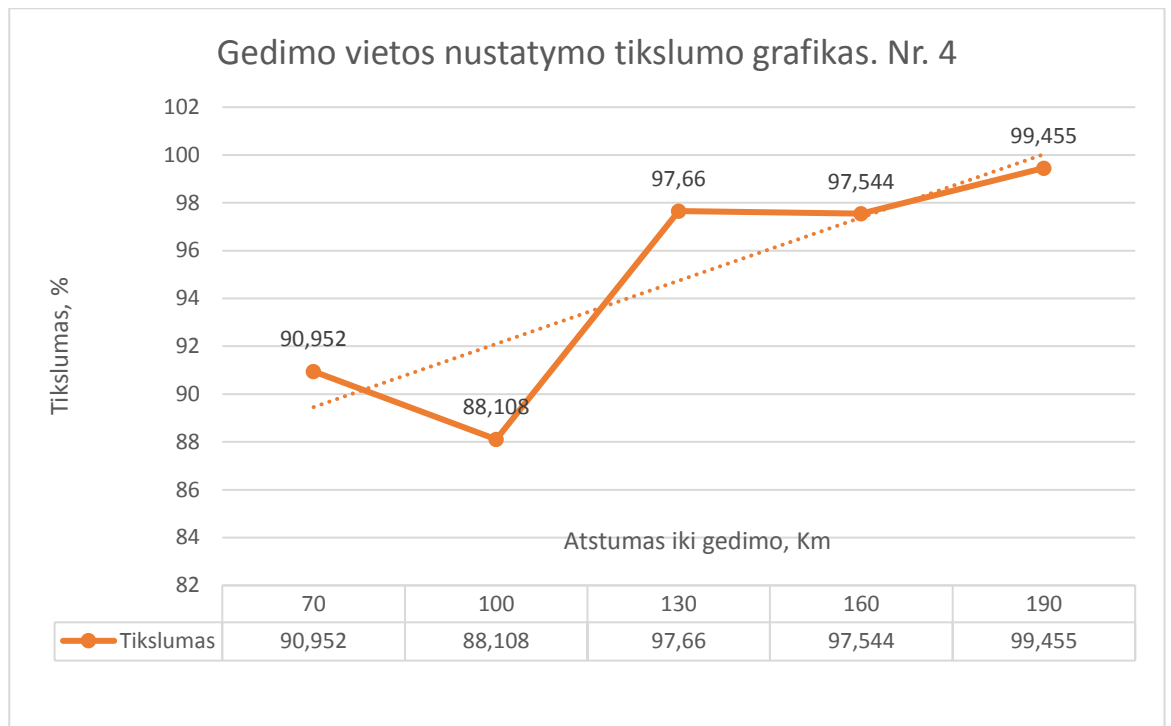
Po lentelę toliau atvaizduoju grafikus, kurie sudaryti iš virš grafikų esančios lentelės duomenų bei rezultatų. Žr. 4.4 lentelė.

4.4. lentelė. Bandymo Nr.4 rezultatų lentelė.

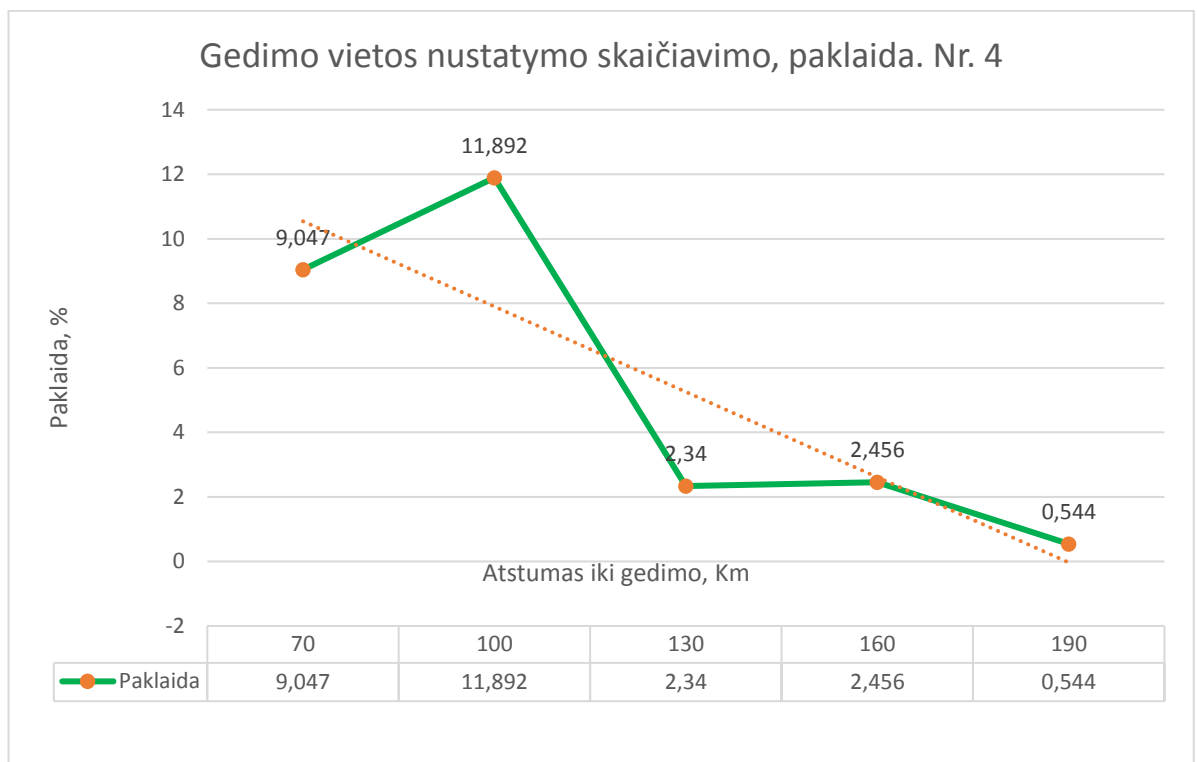
<i>Matavimo Nr.</i>	<i>Zsk , Ω</i>	<i>RRsk , Ω</i>	<i>ZZsk , Ω</i>	<i>Matavimo laikas s.</i>	<i>Atstumas iki gedimo, km.</i>	<i>Tikslumas, %</i>	<i>Paklaida, %</i>	<i>Tikras gedimo vietos atstumas, km.</i>
Matavimas 1	21,98	33,956	50,734	22,63	76,333	90,952	9,0471	70
Matavimas 2	31,4	45,428	80,562	21,69	111,892	88,108	11,892	100
Matavimas 3	40,83	50,221	104,542	24,15	133,041	97,660	2,340	130
Matavimas 4	48,21	57,22	136,25	24,42	163,928	97,544	2,456	160
Matavimas 5	59,67	63,65	177,64	24,52	191,034	99,455	0,544	190
<i>Vidutinis Skaičiavimo greitis</i>	x	x	x	<b>23,482</b>	x	x	x	
<i>Viso vidutinis tikslumas, paklaida</i>	x	x	x	x	x	<b>94,745</b>	<b>5,255</b>	x

Pateikiu gedimo vietos nustatymo algoritmo tikslumo bei skaičiavimo paklaidos grafikus keičiantis gedimo vietos atstumui. Kai pilnutinė ir aktyvioji varža parenkama pagal linijos kampą.





4.1.7 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos, kai varžos parenkamos pagal linijos kampą.



4.1.8 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos, kai varžos parenkamos pagal linijos kampą.

Iš šių grafikų matome, kad gedimo vietos nustatymo algoritmo tikslumas didėja didėjant atstumui iki gedimo vietos. Kas prieš tai buvusiuose bandymuose buvo priešingai, ten didėjant atstumui gedimo vietos algoritmo tikslumas mažėjo.

Punktyrinė linija parodo duomenų tendenciją. Šio algoritmo vidutinis tikslumas, keičiantis atstumui iki gedimo vietos nuo 70 iki 190 km., yra 94,745 %, o šio algoritmo skaičiavimo paklaida yra 5,255%. Šiame bandyme gedimo varža skaičiavime įvertinta, mes ją žinojome. Gedimo varžą nustatome rankiniu būdu „Simulink“ programoje.

Taip pat nustatytas vidutinis skaičiavimo greitis šio bandymo metu yra 23,482 sekundės. Taigi skaičiavimo laikas yra 3 – imis sekundėmis didesnis, nei 1 – ojo bandymo metu. Tačiau šis skaičiavimo metodas skaičiavimus atlieka greičiau beveik minute, nei tada, kai naudojamas DC dedamosios pašalinimas. Tačiau nenaudojant šio filtro tikslumas yra šiek tiek prastesnis. Tikslumas taip pat mažesnis nei skaičiavimose, kai gedimo vieta nustatoma pagal mažiausias varžas gedimo metu.

Skaičiavimo tikslumas šio bandymo metu viršija 94 %, tai galėtume vertinti gerai. Pastebimama, kad šio skaičiavimo paklaida mažėja didėjant atstumui iki gedimo vietos, kas yra įdomu.

#### ***4.1.5. Bandymas Nr.5 gedimas fazė – žemė su DC filtru kai gedimo vietos atstumas skaičiuojamas pagal linijos kampą***

Šiame skyriuje aprašytas tyrimas kai gedimo vietos nustatymo algoritmas naudojamas kartu su nuolatinės srovės dedamosios pašalinimo filtru (DC filtru). Ir algoritmas nustato gedimo vietos atstumą pagal formulę (4.7), tačiau gedimo vieta surandama ne pagal mažiausią gedimo pilnutinę ir aktyviąją varžą, o pagal linijos kampą. Ieškamos mažiausios pilnutinės ir aktyviosios varžos, kurios atitiktų pasirinktą linijos kampą. Taigi varžos nebūtinai gali būti mažiausios. Šiame bandyme gedimo varža yra žinoma. Bandymą atliksiu 5 kartus su skirtingomis atstumų reikšmėmis, kai gedimo vieta yra 70,100,130,160,190 km, atstumuose. Bandymo metu nustatysiu koks yra šio algoritmo tikslumas keičiant atstumus iki gedimo vietos. Taip pat, kaip greitai skaičiuojamas algoritmas (kiek laiko užtrunka skaičiavimai). Bei nustatysiu šio algoritmo paklaidą, kiek procentų jis suklysta nuo tikrosios gedimo vietos. Gedimas vyksta tarp fazės C ir žemės. Skaičiavimo algoritmo metodika pateikta 3.2 skyriuje. Matavimų linijos kampas yra 31,5 laipsnis. Bandymo pritaikymas panašus į 4.1.2. skyriuje aprašytą, tačiau gedimo vieta nustatoma, pilnutinę ir aktyviąją varžas parenkant astižvelgiant į linijos kampą.

Gedimo vietos nustatymo paklaida ir tikslumas procentaliai apskaičiuojami pagal tas pačias metodikas kaip ir bandyme Nr. 1 (4.1.1 skyriuje, formulės 4.8, 4.9 ir 4.10).

Algoritmo skaičiavimo greitis (skaičiavimo laikas) surandamas rankiniu būdu chronometro pagalba.

Toliau pateikiama lentelė, kur surašyta visų matavimų duomenys, bei rezultatai skirtingu matavimu metu.

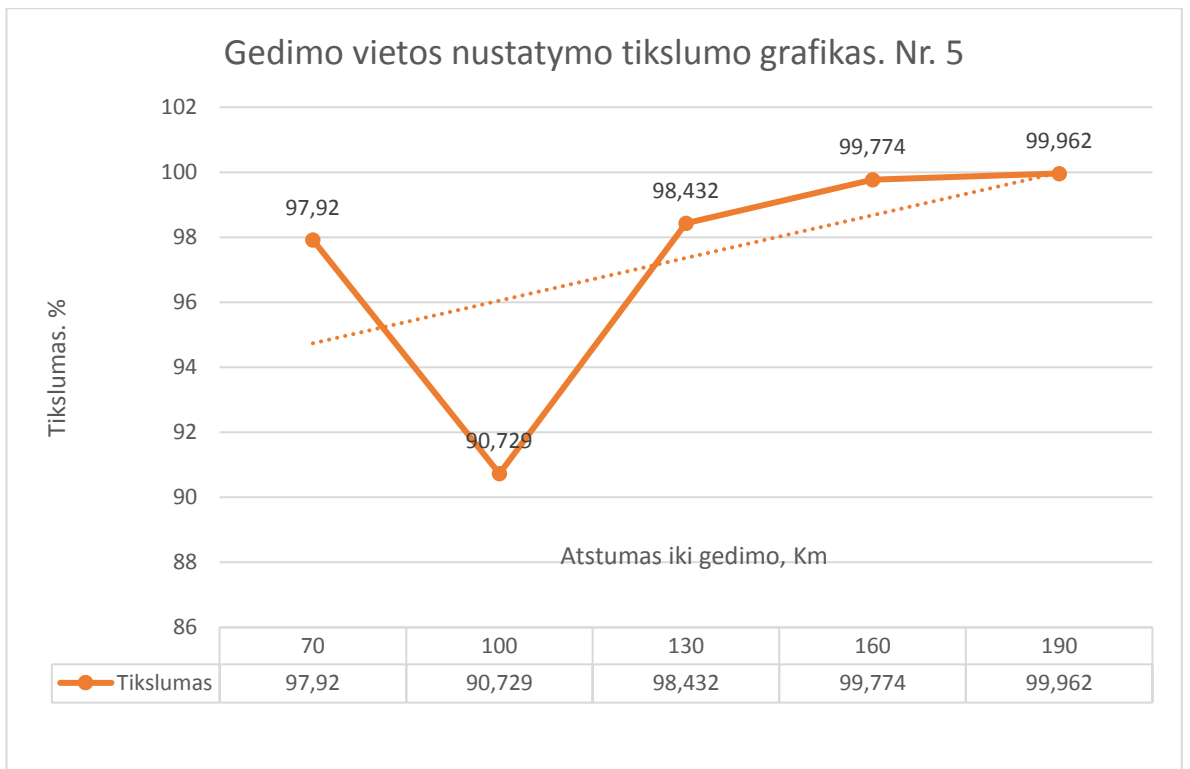
Taip pat, apskaičiuoti algoritmo tikslumo, skaičiavimo greičio ir algoritmo skaičiavimo paklaidos vidurkiai.

Po lentelės toliau atvaizduojami grafikus, kurie sudaryti iš virš grafikų esančios lentelės duomenų bei rezultatų. Žr. 4.5 lentelė.

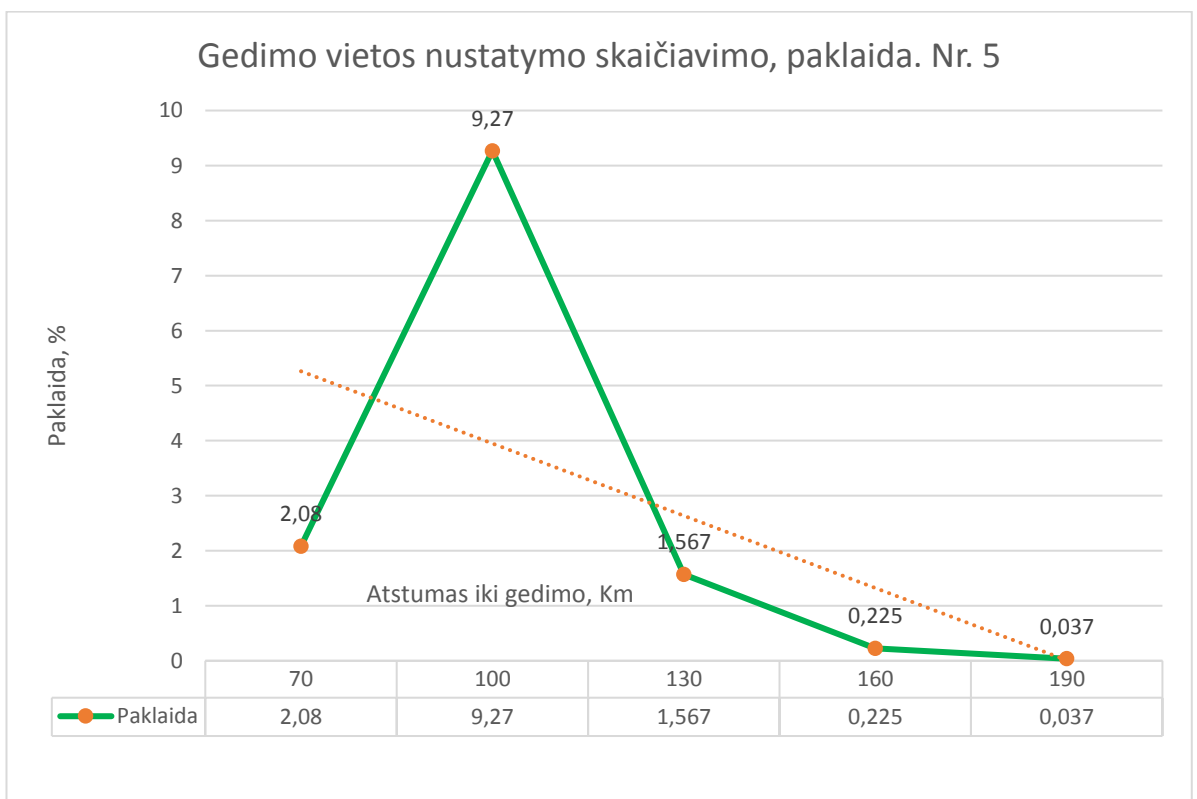
4.5 lentelė. Bandymo Nr.5 rezultatų lentelė.

<i>Matavimo Nr.</i>	<i>Zsk, Ω</i>	<i>RRsk, Ω</i>	<i>ZZsk, Ω</i>	<i>Matavimo laikas s.</i>	<i>Atstumas iki gedimo Km.</i>	<i>Tikslumas, %.</i>	<i>Paklaida, %.</i>	<i>Tikrasis atstumas iki gedimo vietos, km.</i>
Matavimas 1	21,98	33,816	49,522	82,63	71,455	97,92	2,08	70
Matavimas 2	31,4	47,100	81,411	81,63	109,270	90,729	9,27	100
Matavimas 3	40,83	51,431	105,342	80,17	132,037	98,432	1,567	130
Matavimas 4	48,21	57,94	135,25	79,44	160,360	99,774	0,225	160
Matavimas 5	59,67	64,65	177,98	86,48	189,927	99,962	0,037	190
<i>Vidutinis Skaičiavimo greitis</i>	x	x	x	<b>82,07</b>	x	x	x	x
<i>Viso vidutinis tikslumas, paklaida</i>	x	x	x	x	x	<b>97,364</b>	<b>2,635</b>	x

Pateikiu gedimo vietos nustatymo algoritmo tikslumo bei skaičiavimo paklaidos grafikus keičiantis gedimo vietos atstumui. Kai pilnutinė ir aktyvioji varža parenkama pagal linijos kampą. Bei naudojamas DC filtras.



4.1.9 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos, kai varžos parenkamos pagal linijos kampą. Naudojant DC filtrą.



4.1.10 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos, kai varžos parenkamos pagal linijos kampą. Naudojant DC filtrą.

Iš šių grafikų matome, kad gedimo vietos nustatymo algoritmo tikslumas didėja didėjant atstumui iki gedimo vietos. Kas prieš tai buvusiuose 1 – aje, 2 – aje ir 3 – iame bandymuose buvo priešingai, ten didėjant atstumui gedimo vietos algoritmo tikslumas mažėjo. Taip pat pastebime DC filtro įtaką, kuri padidina gedimo vietos nustatymo algoritmo tikslumą ir gedimo vieta surandama beveik 3 procentais tiksliau nei 4 – ajame bandyme.

Punktyrinė linija parodo duomenų tendenciją. Šio algoritmo vidutinis tikslumas, keičiantis atstumui iki gedimo vietos nuo 70 iki 190 km., yra 97,364 %, o šio algoritmo skaičiavimo paklaida yra 2,635%. Penktame bandyme gedimo varža skaičiavime įvertinta, mes ją žinojome. Gedimo varžą nustatome rankiniu būdu „Simulink“ programoje.

Taip pat nustatytas vidutinis skaičiavimo greitis šio bandymo metu yra 82,07 sekundės. Taigi skaičiavimo laikas yra didesnis minute nei bandymuose kai nenaudojamas DC filtras. Šio metodo pranašumas prieš anksčiau atliktus, kad jis yra tikslesnis tada, kai atstumas didėja iki gedimo vietos. Bei skaičiavimo paklaida yra praktiškai vienoda kaip 1 – oje bandymo ir apie procentu mažesnė nei 2 – oje bandymo metu.

Skaičiavimo tikslumas šio bandymo metu viršija 97 %, tai galėtume vertinti tikrai gerai. Bei pastebimama, kad šio skaičiavimo paklaida mažėja didėjant atstumui iki gedimo vietos, kas yra įdomu.

#### **4.1.6. Bandymas Nr.6 gedimas fazė – žemė su DC filtru metodika 2**

Šiame skyriuje aprašytas tyrimas kai gedimo vietos nustatymo algoritmas naudojamas kartu su nuolatinės srovės dedamosios pašalinimo filtru (DC filtru). Algoritmas nustato gedimo vietos atstumą pagal žemiau pateiktą formulę. Bandymą atliksiu 5 kartus su skirtingomis atstumų reikšmėmis, kai gedimo vieta yra 70,100,130,160,190 km, atstumuose. Bandymo metu nustatysiu koks yra šio algoritmo tikslumas keičiant atstumus iki gedimo vietos. Taip pat, kaip greitai skaičiuojamas algoritmas (kiek laiko užtrunka skaičiavimai). Bei nustatysiu šio algoritmo paklaidą, kiek procentų jis suklysta nuo tikrosios gedimo vietos. Gedimas vyksta tarp fazės C ir žemės.

Gedimo vieta skaičiuojama pagal 3.2 ir 3.3 naudojamas metodikas. Gedimo vietos atstumas surandamas pagal 3.2 skyriuje aprašytą algoritmą, kuris skaičiuoja mažiausią pilnutinę ir aktyviają varžas, kurių reikšmes išrenka iš apskaičiuotų varžų vidurkių kreivių. Tada paskaičiuotiems duomenims pritaikomas 3.3 sk. aprašytas filtras ir perskaičiuojamos vidutinės varžų reikšmės. Pagal išfiltruotus duomenis pritaikomas 3.2 sk. aprašytas algoritmas, kuris suranda mažiausias vertes iš vidutinių varžų kreivių.

Bei perskaičiuojama gedimo vieta su jau išfiltruotais duomenimis pagal tokią formulę:

$$kint = \frac{ZZsk}{Zsk} \quad (4.11);$$

Tada gedimo viet1 gautume taip:

$$kint = \frac{ZZsk}{Zsk} = \frac{37,344}{21,98} = 1,69899$$

Gedimo vietos nustatymo paklaida ir tikslumas procentaliai apskaičiuojami pagal tas pačias metodikas kaip ir bandyme Nr. 1 (4.1.1 skyriuje, formulės 4.8, 4.9 ir 4.10).

Algoritmo skaičiavimo greitis (skaičiavimo laikas) surandamas rankiniu būdu chronometro pagalba.

Toliau pateikiama lentelė, kur surašyti visų matavimų duomenys, bei rezultatai gauti skirtingų matavimų metu.

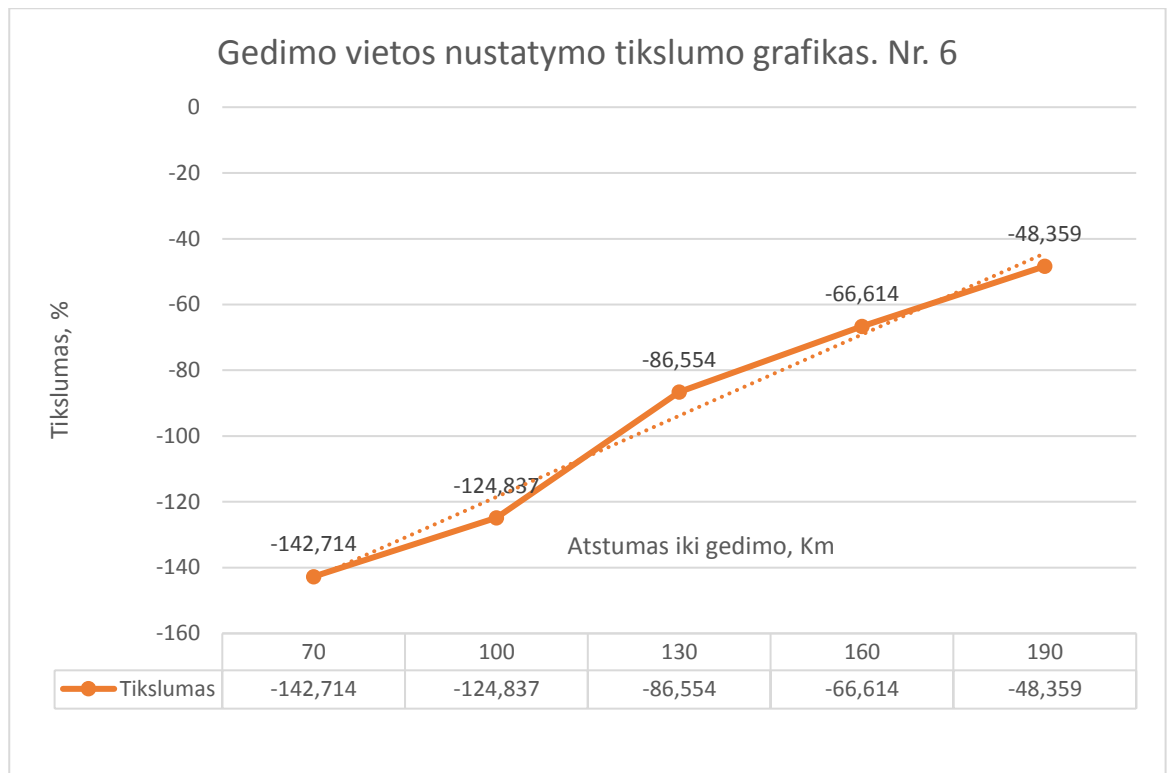
Taip pat, apskaičiuoti algoritmo tikslumo, skaičiavimo greičio ir algoritmo skaičiavimo paklaidos vidurkiai.

4.6. lentelė. Bandymo Nr.6 rezultatų lentelė.

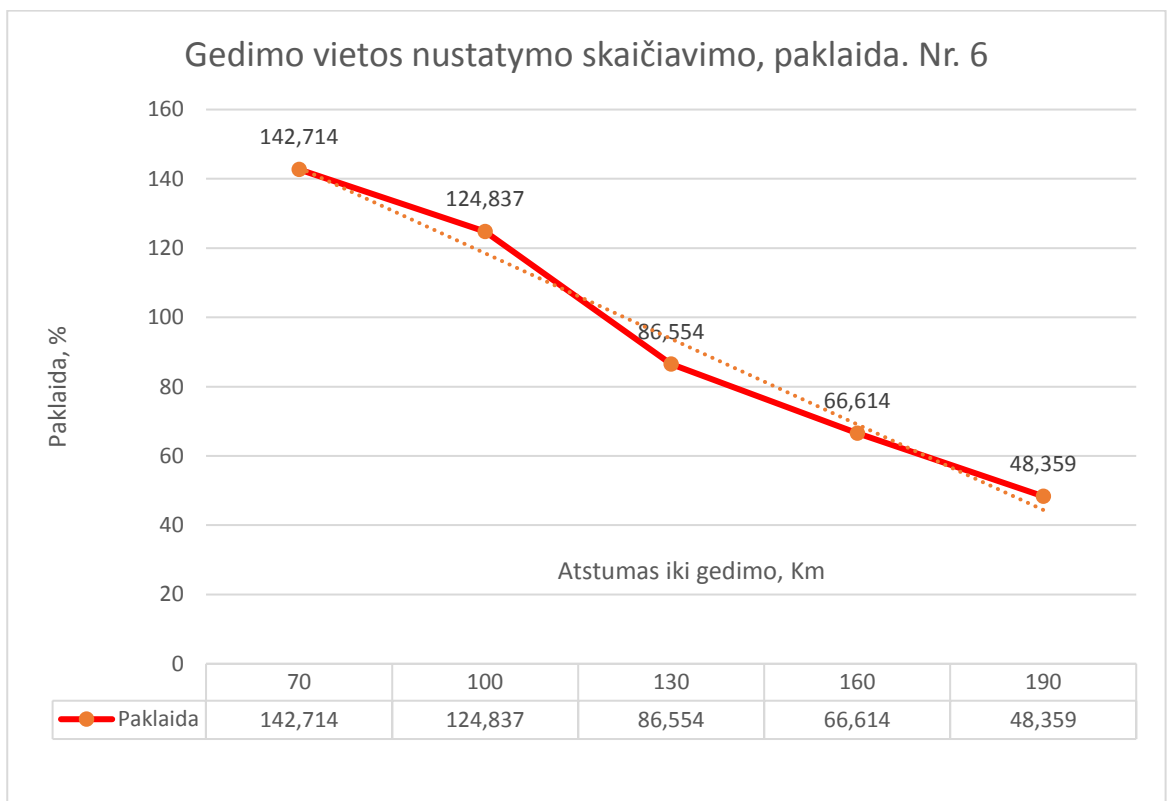
<i>Matavimo Nr.</i>	<i>Zsk, Ω</i>	<i>ZZsk, Ω</i>	<i>Matavimo greitis, s.</i>	<i>Gedimo vietos atstumas, km.</i>	<i>Tikslumas, %.</i>	<i>Paklaida, %.</i>	<i>Tikrasis atstumas iki gedimo, km.</i>
Matavimas 1	21,98	37,344	82,14	169,899	-142,714	142,714	70
Matavimas 2	31,4	70,599	80,87	224.837	-124.837	124.837	100
Matavimas 3	40,83	99,021	81,25	242,520	-86,554	86,554	130
Matavimas 4	48,21	128,52	82,52	266,583	-66,614	66,614	160
Matavimas 5	59,67	178,20	82,66	298,642	-48,359	48,359	190
<i>Vidutinis Skaičiavimo greitis</i>	x	x	<b>81,888</b>	x	x	x	x
<i>Viso vidutinis tikslumas, paklaida</i>	x	x	x	x	<b>-93,815</b>	<b>93,815</b>	x

Po lentele toliau atvaizduoju grafikus, kurie sudaryti iš virš grafikų esančios lentelės duomenų bei rezultatų. Žr. 4.6 lentelė.

Pateikiu gedimo vietos nustatymo algoritmo tikslumo bei skaičiavimo paklaidos grafikus keičiantis gedimo vietos atstumui.



4.1.11 pav. Gedimo vietos nustatymo algoritmo tikslumo grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras. Metodika 2.



4.1.12 pav. Gedimo vietos nustatymo algoritmo skaičiavimo paklaidos grafikas kintant atstumui iki gedimo vietos, kai naudojamas nuolatinės srovės dedamosios šalinimo filtras. Metodika 2.

Iš šių grafikų puikiai matome, kad gedimo vietos nustatymo algoritmo tikslumas labai prastas, jis visiškai netinkamas naudoti su mano pateiktu gedimo vietos nustatymo modeliu.

Punktyrinė linija parodo duomenų tendenciją. Šio algoritmo vidutinis tikslumas, keičiantis atstumui iki gedimo vietos nuo 70 iki 190 km., yra -93,815 %, o šio algoritmo skaičiavimo paklaida yra 93,815 %. Šiame bandyme gedimo varža yra žinoma. Taip pat nustatytas vidutinis skaičiavimo greitis šio bandymo metu yra 81,888 sekundės.

Taigi iš šio bandymo rezultatų matome: nors skaičiavimo greitis panašus į kitų bandymų, kur naudojamas DC filtras, tačiau jo tikslumas visiškai netinkamas. Paklaida viršija 90 %. Taigi šios metodikos daugiau nebetiriame ir nebevertiname, kaip tinkamos naudoti. Tikslumas vertinamas labai blogas.

## **4.2. Gedimo vietos algoritmo tikslumo vertinimas kintant gedimo pobūdžiui**

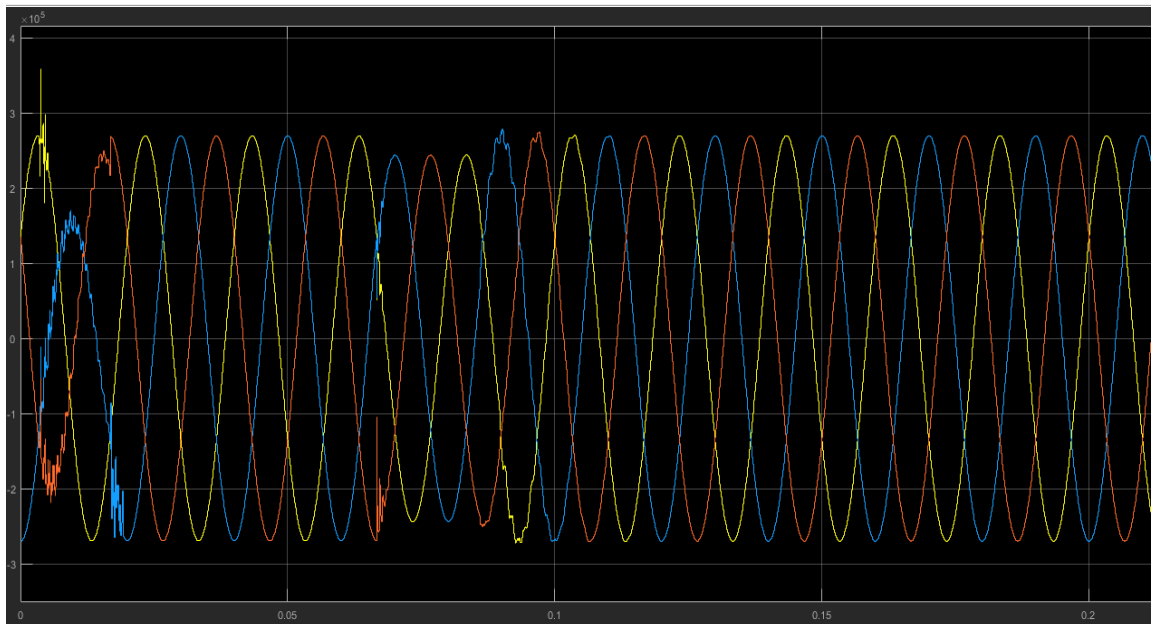
Šiame skyriuje nagrinėsiu kiek mano pateikiamas modelis yra pritaikomas jei gedimas yra fazė – fazė, fazė – fazė – žemė, 3 fazės ir 3 fazės – žemė. Tai atliksiu sumodeliuojant sistemą su Simulink programos pagalba, bei imituojant šiuos gedimus. Vienos fazės-žemės gedimo nevertinsiu šiame skyriuje jį jau įvertinau 4.1 skyriuje.

### **4.2.1. Gedimas fazė-fazė-žemė bei fazė-fazė**

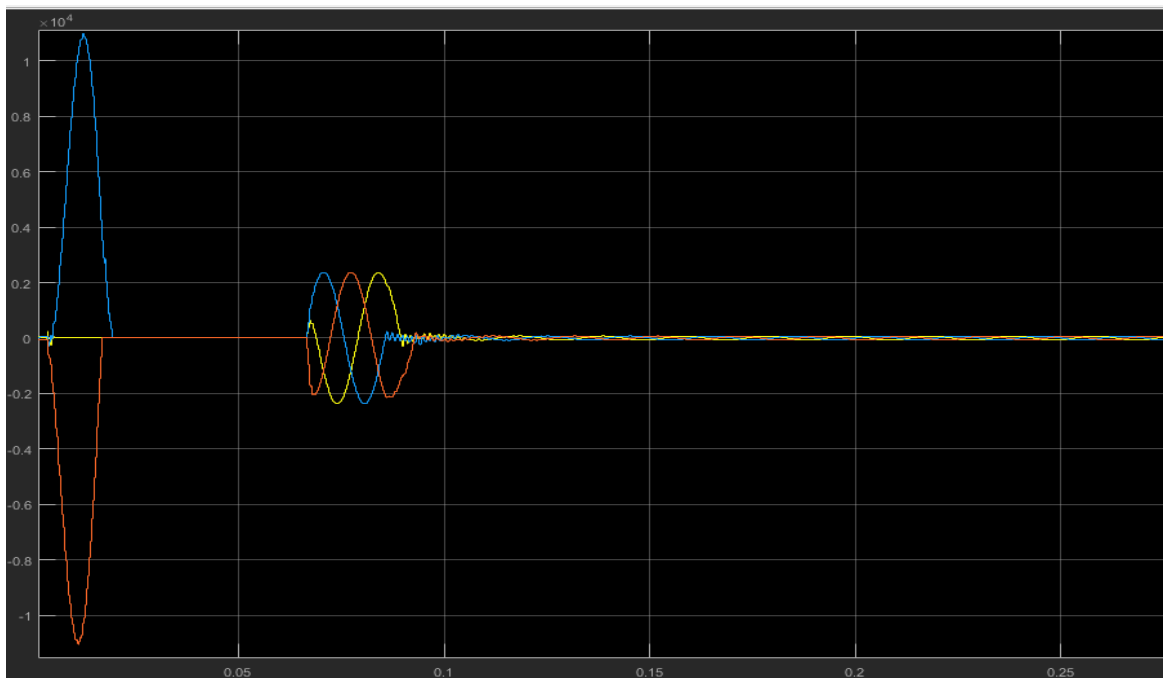
Šiame skyriuje trumpai aprašysiu kokios yra pateikiamo modelio galimybės esant gedimams fazė – fazė – žemė ir fazė – fazė.

Imituojant gedimus atvaizduojamos tokios itampų, srovių, gedimo metu esančių apskaičiuotų pilnutinės,aktyviosios ir reaktyviosios varžų kreivės. Taip pat atvaizduojami varžų fazių kampų duomenys. Pateikiamos fazė – fazė – žemė gedimo metu esančios pereinamųjų procesų kreivės.

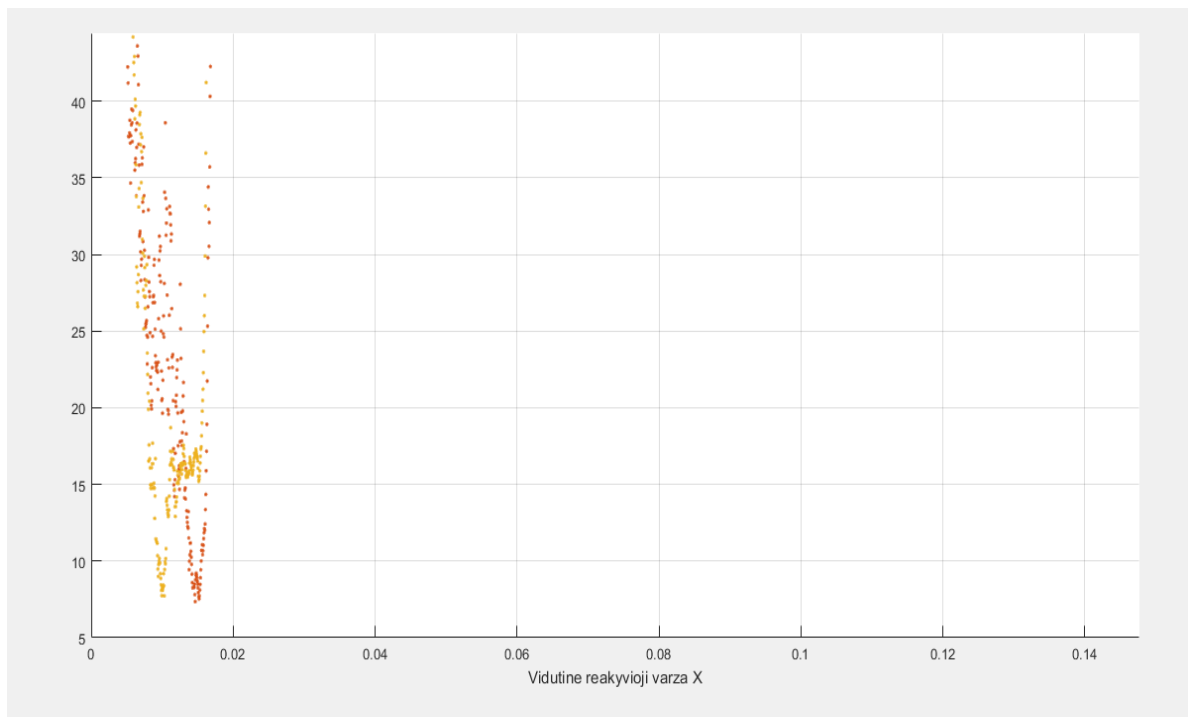




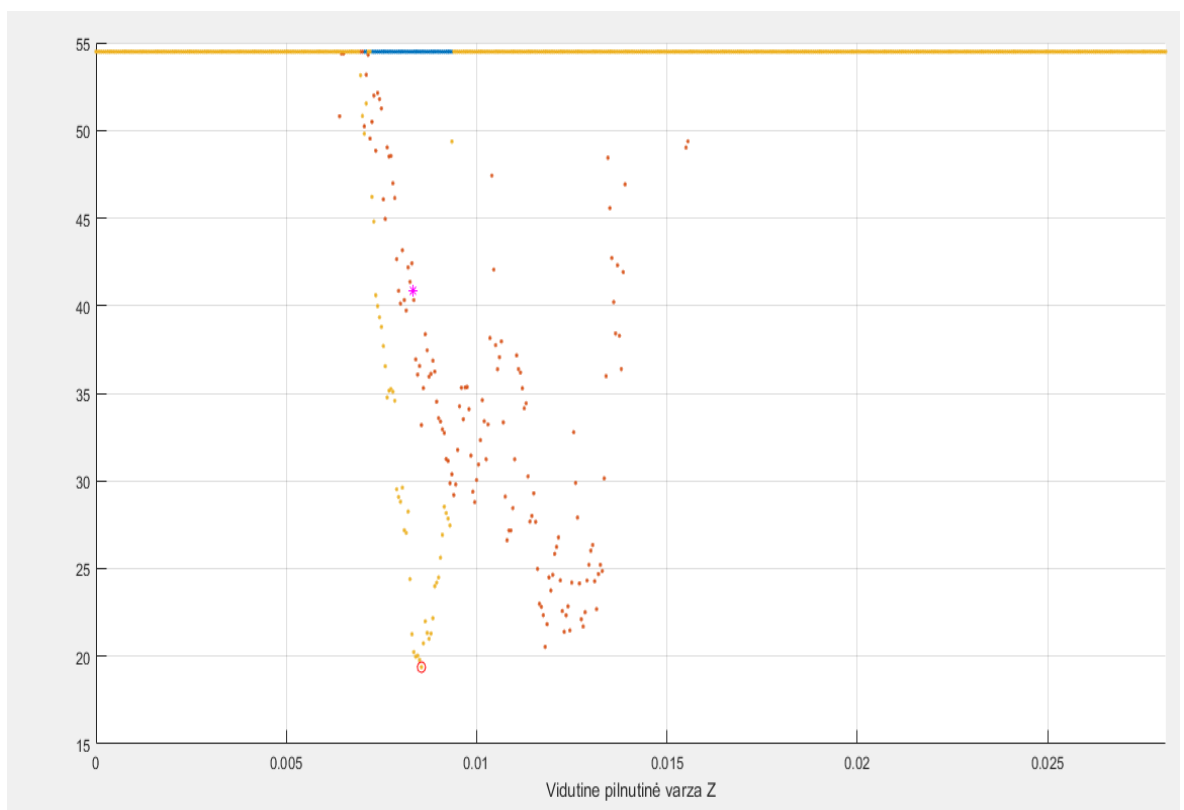
4.2.1 pav. Modeliuojamo gedimo fazė – fazė – žemė įtampos kitimo kreivė Simulink programoje.



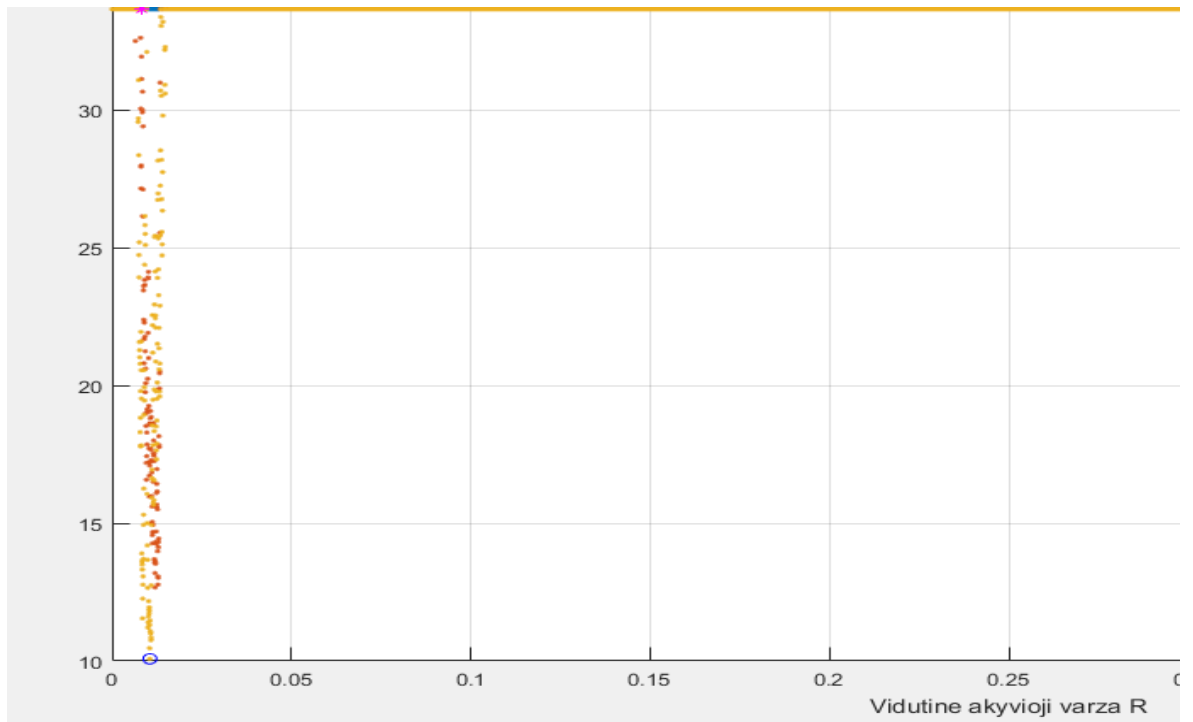
4.2.2 pav. Modeliuojamo gedimo fazė – fazė – žemė srovės kitimo kreivė Simulink programoje.



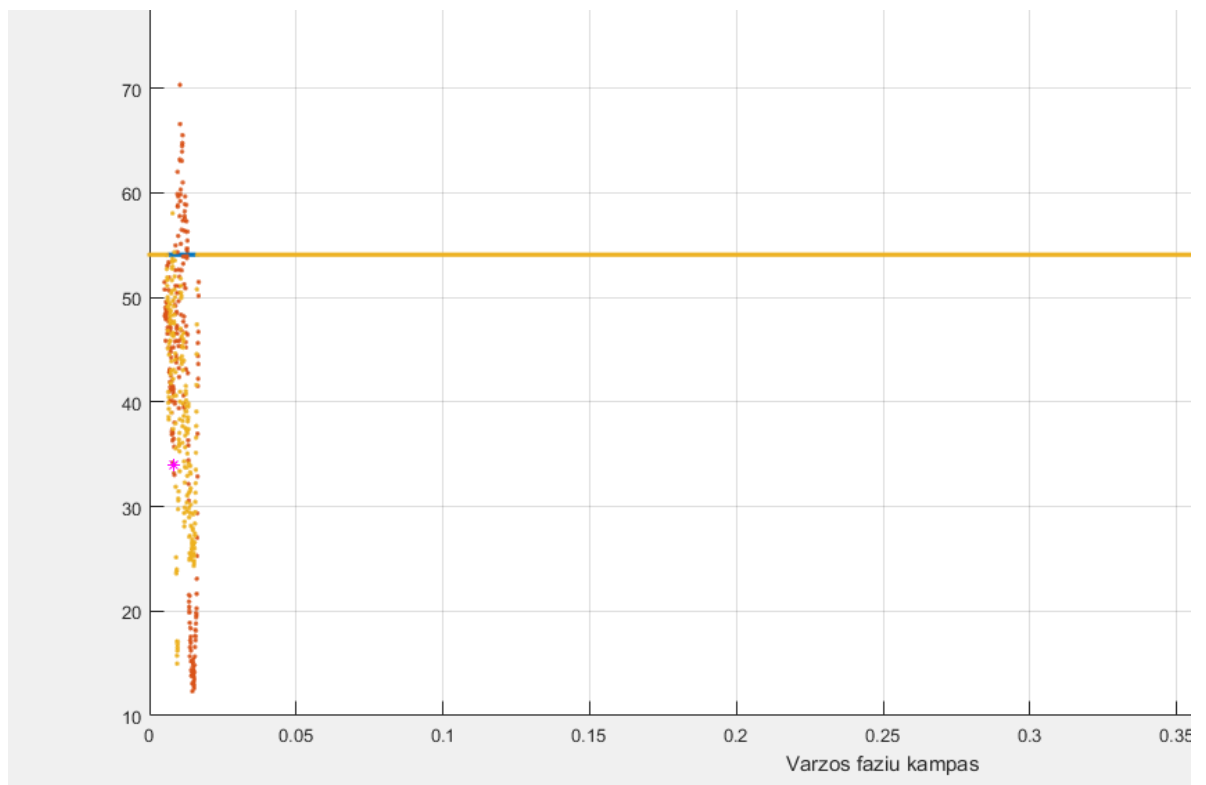
4.2.3 pav. Modeliuojamo gedimo fazė – fazė – žemė vidutinės reaktyviosios varžos gedimo metu kitimo kreivė.



4.2.4 pav. Modeliuojamo gedimo fazė – fazė – žemė vidutinės pilnutinės varžos gedimo metu kitimo kreivė.



4.2.5 pav. Modeliuojamo gedimo fazė – fazė – žemė vidutinės pilnutinės varžos gedimo metu kitimo kreivė.



4.2.6 pav. Modeliuojamo gedimo fazė – fazė – žemė, varžų fazių kampai gedimo metu.

Taigi, mano pateikiamas modelis gali atvaizduoti ir kitus gedimus ne tik vienos fazės su žeme gedimo, tokius kaip fazė – fazė bei fazė – fazė – žemė duomenis, bei paskaičiuoti varžas. Tačiau jis

negali išskaičiuoti gedimo vietos atstumo. Tai galima padaryt nebent rankiniu būdu. Jeigu vertiname gedimus tarp 3 fazių arba 3 fazių ir žemės. Mano pateikiamas modelis negali atvaizduoti varžų, per sudėtingi skaičiavimai šiam modeliui.

Jeigu bandytume paskaičiuoti gedimo vietos atstumą ranka iš turimų kreivių, gedimo vietos nustatymo tikslumas būtų tarp 77 – 88 % atsižvelgiant kokią įtaką turėtų gedimo varža.

Taigi mano pateikiamas modelis negali savarankiškai išskaičiuoti gedimo vietos, jis gali tik pateikti duomenis, kad rankiniu būdu galėtume išsiskaičiuoti atstumą. Tačiau skaičiuojant rankiniu būdu tikslumas nebūtų labai geras. Todėl šis modelis geriausiai tinkamas naudoti kai yra vienfaziai trumpieji jungimai tinkle. Tada šio modelio tikslumas yra labai geras, jei žinoma gedimo varža bei jis pats suranda gedimo vietą.

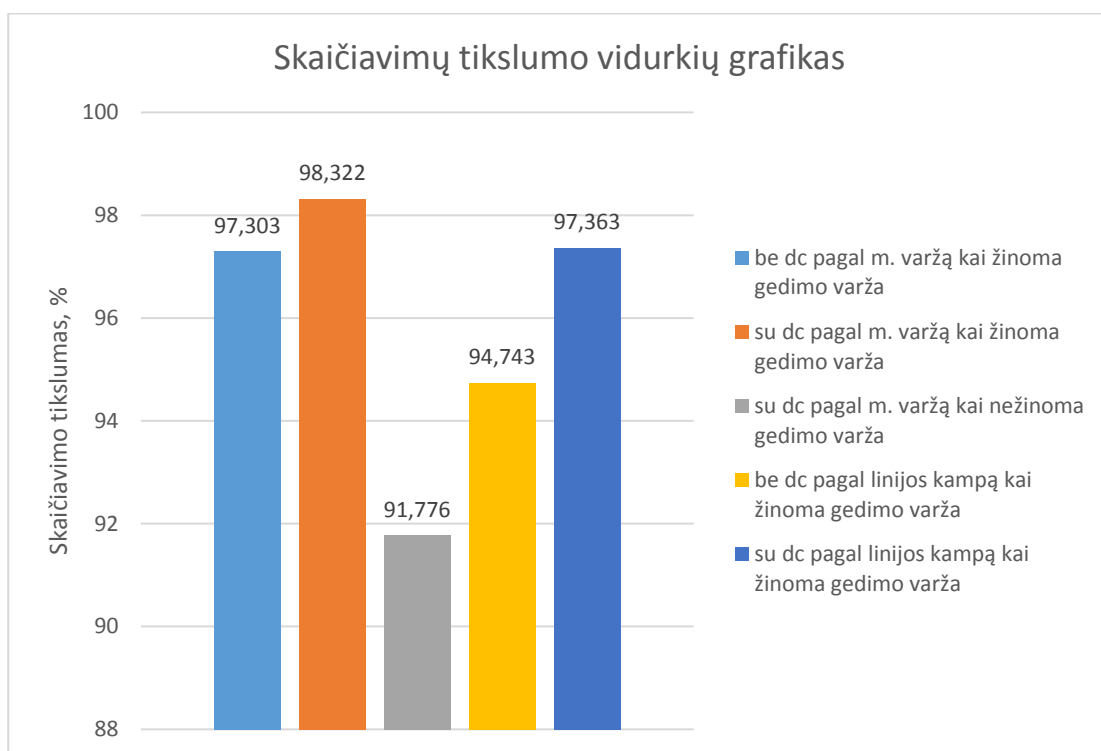
## 5. GEDIMO VIETOS NUSTATYMO ALGORITMO MODELIO ANALIZĖ

Šiame skyriuje pateiksiu trumpą mano pateikiamo gedimo vietos nustatymo modelio įvertinimą pagal atliktus bandymus ir gautus rezultatus. Apibendrinsiu šio modelio galimybes. Bei gautus rezultatus.

### 5.1. Modelio tikslumas

Čia pateikiu mano sudarytą grafiką (5.1.1 pav.) pagal bandymų metu gautus rezultatus. Pateikiu vidutinius bandymų naudotų skaičiavimų metodikų rezultatus ir juos palyginu.

Grafikuose atvaizduotas bandymų metu nustatytas vidutinis tikslumas.



5.1.1 pav. Bandymų metu nustatytų skaičiavimų tikslumo vidurkių grafikas.

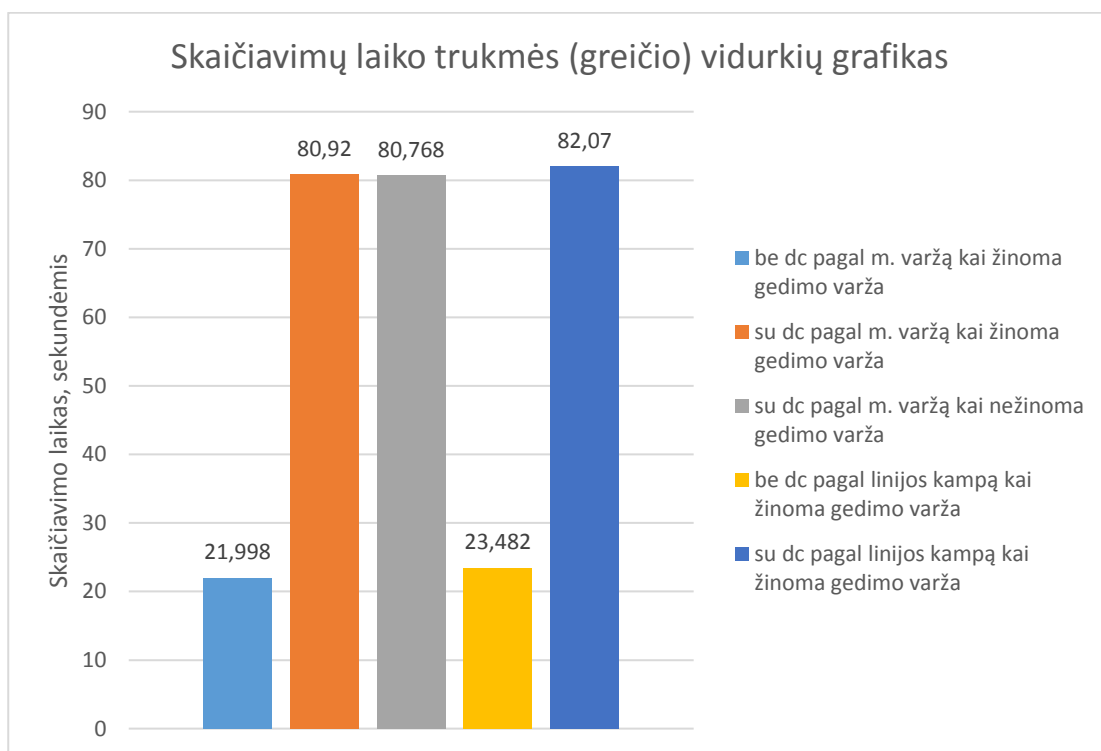
5.1.1 paveikslėlyje pateikiamas vidutinis tikslumas gautas skaičiavimų metu atliekant bandymus aprašytus 4.1 skyriuje. Iš pateikto grafiko matome, kad tiksliausiai modelis skaičiuoja gedimo vietos atstumą, kai naudojama metodika yra paremta mažiausių varžų nustatymu gedimo metu, bei šiai metodikai pritaikytas nuolatinės srovės dedamosios filtras. Jis pašalina DC dedamąją taip pašalindamas nereikalingus triukšmus, bei pagerinant skaičiavimų tikslumą šiame modelyje. Tikslumas siekia 98,322 % kai žinoma gedimo varža (bandymas Nr.2.). Pastaba: šis modelis tikslesnis kuomet atstumas mažesnis, didėjant atstumui tikslumas pamažu mažėja.

Taip pat tiksliai gedimo vietą suranda ir kitos dvi naudojamos metodikos. Kai nenaudojamas DC filtras ir ieškoma mažiausių varžų, tikslumas siekia 97,303 % (bandymas Nr.1). Bei kai naudojamas DC filtras, tačiau gedimo vieta surandama pagal linijos kampą. Tuo metu varžos parenkamos pagal tai, o ne kaip pirmuose trijuose bandymuose. Šio skaičiavimo tikslumas siekia 97,363 %, bei kai gedimo varža žinoma (bandymas Nr. 5). Jis išsiskiria iš kitų metodikų, kad jo tikslumas gerėja, kai atstumas iki gedimo vietos didėja. Tai yra įdomu. Mažiausiai tikslios skaičiavimų metodikos yra kai nenaudojamas DC filtras ir gedimo vieta nustatinėjama pagal linijos kampą. Tada tikslumas siekia 94,743 % (bandymas Nr.4). Bei kai yra nežinoma gedimo varža ir atstumas iki gedimo vietos surandamas pagal mažiausias varžas gedimo metu, pritaikant DC filtrą. Šio metodo tikslumas siekia 91,776 %.

Šie bandymai atlikti esant vienos fazės su žeme gedimams juos simuliuojant Simulink programos pagalba. Rezultatus vertinu teigiamai, nes buvo surastos metodikos, kurios esant vienfaziam gedimui, gedimo vietą suranda labai tiksliai jei yra žinoma gedimo varža. Bandymai atlikti modeliuojant aukštos įtampos tinklus bei su realiais gedimų duomenimis 110 – 330 kV linijose.

## 5.2. Modelio greitis

Grafike (5.2.1 pav.) pateikiama bandymų metu nustatytas vidutinis skaičiavimo greitis skirtingų bandymų metu.



5.2.1 pav. Bandymų metu nustatytų skaičiavimų laiko trukmės (greičio) vidurkių grafikas.

Kaip matome iš 5.2.1 pav. grafiko greičiausia naudojama metodika su mano pateiktu modeliu yra kai skaičiavimai atliekami be DC filtro ir ieškamos mažiausios varžos gedimo metu. Šio metodo vidutinis skaičiavimų greitis yra 21,998 sekundės. (bandymo Nr.1). Toliau antra pagal mažiausią skaičiavimo laiką metodika, kurią naudoja mano pateiktas modelis, yra kai nėra naudojamas DC filtras, tačiau gedimo vieta apskaičiuojama ieškant mažiausios varžos pagal pasirinktą linijos kampą. Šio metodo skaičiavimų laikas užtrunka vidutiniškai 23,482 sekundės. (bandymo Nr. 4). Ilgiausiai skaičiavimai vidutiniškai atliekami tada, kai naudojamas DC filtras, tai prailgina skaičiavimų trukmę beveik viena minute. Greičiausia metodika, kai naudojamas DC filtras yra tada, kai ieškoma mažiausia varža (pilnutinė, aktyvioji) ir tai trunka vidutiniškai 80,768 sekundės, (bandymo 3 metu, kai gedimo varža nežinoma) ir 80,92 sekundės (bandymo 2 metu, kai gedimo varža žinoma). Lėčiausiai skaičiavimai vyksta su mano pateikiamu modeliu, kai naudojamas DC filtras bei ieškamos varžos pagal pasirinktą linijos kampą. Šio metodo greitis yra vidutiniškai 82,07 sekundės. Taigi iš atliktų bandymų matome, kad skaičiavimai trunka tikrai neilgai. Jei naudojamas DC filtras tai trunka ~80 sekundžių ir labai greitai kai nenaudojamas DC filtras. Tada skaičiavimai trunka maždaug ~23 sekundes. Skaičiavimų greitis taip pat priklauso nuo duomenų kiekio bei kokio pajėgumo yra naudojamas asmeninis kompiuteris. Šie bandymai buvo atlikti naudojant šiek tiek daugiau nei 10000 duomenų bei naudojant asmeninį kompiuterį pagamintą prieš 6 -erius metus, kuris nėra labai galingas. Taigi turint šiuolaikinį asmeninį kompiuterį skaičiavimai būtų atliekami gerokai greičiau su tokia duomenų imtimi.

### **5.3. Modelio skaičiavimų paklaida**

Šiame skyriuje pateikiu sudarytą grafiką (5.3.1 pav.) pagal bandymų metu gautus rezultatus. Atvaizduoju vidutinius bandymų naudotų skaičiavimų metodikų rezultatus ir juos palyginu.

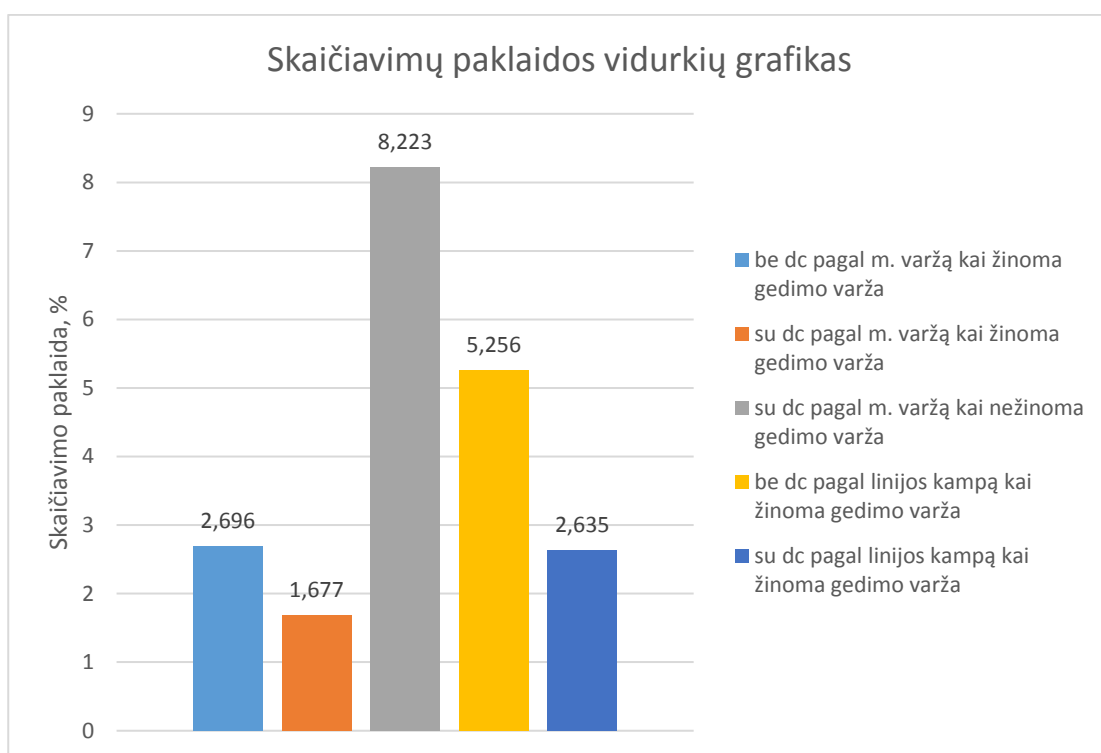
Grafikuose pateikiamos bandymų metu nustatytos vidutinės skaičiavimų paklaidos.

Kaip matome iš žemiau pateikto grafiko 5.3.1 pav., mažiausia skaičiavimų paklaida naudojant mano pateiktą modelį yra tada, kai naudojamas DC filtras ir pašalinamos nuolatinės srovės dedamosios iš naudojamų duomenų. Bei ieškamos mažiausios gedimo metu esančios pilnutinė bei aktyvioji varžos. Šios metodikos paklaida yra 1,677 %. Ir ji yra mažiausia iš visų mano tirtų skaičiavimų metodikų su šiuo modeliu (bandymo Nr.1). Tačiau gedimo varža turi būti žinoma ar pašalinama kokiu tai būdu. Kiti du metodai, kurie turi mažiausią skaičiavimų paklaidą, yra bandymo Nr. 5, kai naudojamas DC filtras, o skaičiuojamos mažiausios varžos surandamos pagal pasirinktą linijos kampą gedimo metu. Šio metodo paklaida 2,635%. Bei kitas metodas tai yra bandymo Nr.1, kai atliekami skaičiavimai naudojant DC filtrą bei ieškamos mažiausios varžos esančios gedimo metu

(aktyvioji, pilnutinė). Šio skaičiavimo metodo paklaida yra 2,696 %. Šiuose dviejuose bandymuose gedimo varžos žinomos.

Nr. 4 bandymo metu atlikti skaičiavimai parodė 5,256 % paklaidą. Šis metodas nenaudoja DC filtro, bei mažiausios varžos surandamos pagal linijos pasirinktą kampą. Pastebima, kad Nr.4 bandymo metodika ir Nr. 5 bandymo metodika parodė, kad atliekant skaičiavimus pagal linijos pasirinktą kampą, šių metodikų paklaida mažėja didėjant atstumui iki gedimo vietos. Priešingą tendenciją parodė Nr. 1 bandymo metodika bei bandymo Nr.2 metodika. Tada paklaida mažėdavo mažėjant atstumui iki gedimo vietos.

Pati didžiausią paklaidą darė skaičiavimas atliktas Nr.3 bandymo metu. Kai buvo naudojamas DC filtras ir ieškoma mažiausios varžos, tačiau tuo metu nebuvo žinoma gedimo varža. Ši skaičiavimo paklaida siekia 8,223 % ir buvo didžiausia iš šių penkių metodikų.



5.3.1 pav. Bandymų metu nustatytų skaičiavimų paklaidų vidurkių grafikas.

Pažymėtina, kad norint, kad skaičiavimų metodikų paklaidos būtų mažesnės bei jų tikslumas būtų didesnis, reikia žinoti kokia yra gedimo varža. Kad žinotume jos įtaką skaičiavimams. Arba reikia tobulinti šį modelį ir rasti būdą kuris šalintų gedimo varžą iš skaičiavimuose naudojamų duomenų ar iš gautų rezultatų.

Šios metodikos tampa labai tikslios, kai gedimo varža žinoma. Šis modelis tiksliausiai pritaikomas vienfazio gedimo metu.



## 6. ALGORITMO PRIVALUMAI, TRŪKUMAI BEI PRITAIKYMAS ELEKTROS ENERGETIKOS SISTEMOJE

Šiame skyriuje apžvelgsiu visame darbe aptarinėjamo gedimo vietos nustatymo algoritmo kaip platformos panaudojimą elektros energetikos sistemoje. Taip pat šio algoritmo galimybes.

### 6.1. Algoritmo privalumai ir trūkumai

#### *Privalumai*

- Iš atliktų bandymų nustatėme, kad šis modelis bei pritaikomi algoritmai gedimo vietai surasti, tokie kaip mažiausios vidutinės varžos suradimas kartu naudojant DC filtrą su duomenimis, yra labai tikslus. Jo tikslumas viršija 98,3%.
- Šis modelis gali būti pritaikomas su realiais duomenimis.
- Šis modelis nustato gedimo vietą esant fazė – žemė gedimui labai tiksliai.
- Šis modelis taip pat gali atvaizduoti vidutines varžas esant gedimams tokiems kaip fazė-fazė ar fazė – fazė – žemė
- Pagal atvaizduotas vidutines varžas galima išskaičiuoti atstumą iki gedimo vietos pakankamai tiksliai.
- Šis modelis atvaizduoja gedimų metu esančius pereinamuosius procesus. Parodo kaip per tam tikrą laiko momentą gedimo metu keičiasi visų fazių srovių kreivės bei įtampų kreivės.
- Šis modelis atvaizduoja gedimo metu esančias vidutines varžas (aktyviąją, reaktyviąją ir pilnutinę). Pagal jas galima nustatyti gedimo vietą.
- Šiame modelyje tirtų 3- jų algoritmų tikslumas viršija 97,3 %. (Bandymai Nr. 1, 2 ir 5).
- Pritaikant algoritmus su šiuo modeliu, kai ieškoma mažiausių vidutinių varžų linijoje gedimo metu, pastebėta, kad tikslumas didėja esant mažiam atstumui iki gedimo.
- Pritaikant algoritmus su šiuo modeliu, kai ieškoma mažiausių varžų pagal pasirinktą linijos kampą, tikslumas didėja, kai atstumas iki gedimo vietos yra didesnis. Atstumui viršijus 150 km. šio algoritmo tikslumas siekia 99 %. Paklaida tik 1 %.
- Šie algoritmai yra labai greiti . Greičiausias yra algoritmas, kuris ieško mažiausių vidutinių varžų ir pagal tai nustatoma gedimo vieta. Šio algoritmo skaičiavimo laikas trunka tik apie 20 s.
- Taip pat patobulintas modelis kai pritaikomas nuolatinės srovės dedamosios šalinimas iš skaičiavimo duomenų. Jis ženkliai pagerina algoritmo tikslumą, bei puikiai pašalina

triukšmus. Bandymų metu nustatyta, kad šis filtras sumažina skaičiavimų paklaidą iki 3 – 4 %.

- Kai naudojamas modelis, kur pritaikytas DC filtras, algoritmų skaičiavimo greitis yra pakankamai trumpas ir siekia 80 s.
- Šis modelis gali būti naudojamas su realiais duomenimis, esant tikriems gedimams sistemoje, bet kuriuo metu, kai prižiūrintis personalas yra pasirengęs nustatyti gedimo vietą ir kada yra gaunami gedimo duomenys.
- Šis modelis nereikalauja didelių resursų iš jį pritaikančio asmens asmeninio kompiuterio. Modelis neužima daug vietos kietajame diske (asmeniniame kompiuteryje).
- Šis modelis pritaiko duomenis pagal IEEE C37.111 – 2013 standarto reikalavimus.
- Šio modelio nustatyti gedimo vietos atstumai bei skaičiavimo duomenys gali būti naudojami analizės tikslams. Bei gali būti panaudoti tyrimams tarp skirtingų individų.
- Šis modelis gali būti laisvai pasidalintas tarp skirtingų individų ir pritaikytas su iš skaitmeninių relių gautais gedimo duomenimis.
- Šis modelis pritaikomas su „MATLAB“ programine įranga.
- Šis modelis kaip platforma gali būti naudojamas pritaikant ir kitus gedimo vietos nustatymo algoritmus (pagal pasirinktą metodiką).
- Šis modelis gali būti naudojamas kaip platforma gedimo vietai nustatyti elektros energetikos sistemoje.

### ***Trūkumai***

- Šio modelio trūkumas, kad jo greitį įtakoja tiriamų duomenų kiekis (tačiau gedimui surasti skaičiavimai vis tiek trunka pakankamai greitai iki 2 min. jei naudojama virš 25 tūkst duomenų).
- Šis modelis yra labai tikslus kada žinoma gedimo vieta.
- Šis modelis gali atvaizduoti vidutines varžas, tačiau gedimo vietos nenustato. Esant gedimams tokiems kaip fazė – fazė ar fazė – fazė – žemė.
- Esant gedimams tokiems kaip fazė – fazė ar fazė – fazė – žemė, gedimo vietą reikia išskaičiuoti rankiniu būdu. Tokio skaičiavimo metodo tikslumas siekia iki 80 – 90 %. Tai vidutinis tikslumas.
- Šio modelio paklaida didelė, kai nežinoma gedimo varža. Ir ji gali siekti iki 8,5 % kai kartu naudojamas DC filtras, bet gedimo varža nežinoma.

- Šis modelis nėra pajėgus surasti gedimo vietą, jei vyksta gedimai tokie kaip 3 fazės ar 3 fazės – žemė. Šiam algoritmui vykstantys procesai šių gedimų metu yra per sudėtingi apskaičiuoti gedimo vietą. Jis gali tik atvaizduoti pereinamuosius procesus.
- Modelio tikslumas kai nežinoma gedimo varža ar ji nėra pašalinta siekia 92 %. Todėl, kad pagerinti šios platformos tikslumą, reikia pašalinti gedimo varžą arba ją žinoti.
- Naudojant nuolatinės srovės dedamosios šalinimo filtrą algoritmų skaičivimų laikas padidėja viena minute.
- Šis modelis yra tobulintinas.

## 6.2. Galimybės pritaikyti elektros energetikos sistemoje

Bandymų metu ištyrime šio modelio kaip platformos galimybes pritaikyti elektros energetikos sistemoje. Šis modelis naudoja IEEE C37.111 – 2013 standarto reikalavimus ir pagal gautus duomenis iš skaitmeninių relių gali apskaičiuoti gedimo vietą labai tiksliai, tačiau tam turi būti žinoma gedimo varža arba ji turi būti pašalinama. Bandymuose nustatyti algoritmų tikslumai rodo, kad tiksliausiai skaičiuojama gedimo vieta, kai ieškoma mažiausių vidutinių varžų gedimo metu. Bei naudojamas DC filtras. Šio algoritmo skaičiavimo paklaida šiek tiek viršija 1,5 %. Algoritmo skaičiavimo greitis, kai naudojama daugiau nei 10 tūkst. duomenų yra ~80 s. Tai yra pakankamai greitai. Šis modelis pritaikomas tarp skirtingų individų. Pritaikymui naudojama „MATLAB“ programa. Modelis nereikalauja daug asmeninio kompiuterio resursų. Gedimos vietos galima ieškoti iš karto kai tik yra gaunami duomenys apie gedimą iš skaitmeninės relės. Šie rezultatai apie gedimo vietos apskaičiuotą atstumą gali būti lengvai perduoti dispečeriams. Dispečeriai esantys dispečerinėje gali patys surasti gedimo vietą pasinaudojant šia platforma, jei turi galimybę gauti duomenis iš skaitmeninės relės. Šis modelis labiausiai tinkamas naudoti esant gedimams fazė – žemė. Šio modelio tikslumas šių gedimų metu didžiausias. Taip pat, esant gedimams fazė – fazė ar fazė – fazė – žemė šią platformą galima naudoti gedimo atstumui surasti, tačiau modelis tik atsvaizduoja vidutines varžas linijoje gedimo metu. Gedimo vietą tokiu atveju reikia išskaičiuoti rankiniu būdu. Šio algoritmo pritaikymas daug ekonominių resursų nereikalauja ir gali būti pritaikytas bet kuriam asmeniniam nešiojamajam kompiuteriui kuriame yra galimybė naudoti „MATLAB“ programą. Šis algoritmas yra tobulintinas todėl tikslumas gali būti pagerintas, siekiant tiksliau nustatyti gedimo vietą ištikus avarijai elektros energetikos sistemoje. Ši platforma leidžia naudoti gedimo duomenis analizės tikslams.

## IŠVADOS

1. Sukurta platforma yra pagal IEEE C37.111 – 2013 standarto reikalavimus, tai leidžia gautus rezultatus analizuoti tarp skirtingų individų. Taip pat platforma nereikalauja didelių resursų.
2. Modelis gali surasti gedimo vietą iš vidutinių varžų kreivių pagal pasirinktą linijos kampą. Tyrimo metu nustatyta, kad naudojant šį metodą (Bandymas Nr. 4), tikslumas esant gedimui fazė – žemė yra 94,743 %. Šio metodo skaičiavimo greitis trunka 23,482 sekundes. Iš šio bandymo matome, kad jis yra pakankamai greitas bei pakankamai tikslus. Paklaida siekia 5,256 %.  
Modelis patobulintas naudojant nuolatinės srovės dedamosios šalinimą iš skaičiavimo duomenų. Pritaikius patobulinimą šis metodas pagerina skaičiavimų tikslumą nuo 1,5 iki 4 %.
3. Bandymų metu nustatyta, kad tiksliausias gedimo vietos nustatymo metodas yra kai ieškoma vidutinių mažiausių varžų gedimo metu kartu panaudojant nuolatinės srovės dedamosios šalinimo algoritmą. Šio metodo tikslumas siekia 98,322 %. Skaičiavimo modelio greitis yra 80,92 sekundės. Metodo paklaida siekia 1,677 %. Šį metodą rekomenduočiau naudoti ieškant gedimos vietos su realiais duomenimis, kadangi tai tiksliausias metodas iš mano pritaikytų. Taip pat atsižvelgiant, kad skaičiavimai užtrunka ilgiau jis yra pakankamai greitas ir efektyviai nustato gedimo vietą.
4. Bandymų Nr.1, Nr.2 ir Nr.3 metu nustatyta, kad naudojant metodiką, kai ieškoma gedimo vieta pagal vidutines mažiausias varžas gedimo metu, tikslumas didėja mažėjant tiriamam atstumui iki gedimo vietos. Tai puikiai parodo 4.1. skyriuje pateikti bandymai bei jų tikslumo ir skaičiavimo paklaidos grafikai.
5. Bandymų Nr.4 ir Nr.5 metu nustatyta, kad naudojant metodiką, kai ieškomos vidutinės mažiausios varžos pagal pasirinktą linijos kampą gedimo metu, šio skaičiavimo metodo paklaida mažėja kai yra didinamas atstumas iki gedimo vietos. Kas yra priešingai nei 4-oje išvadoje pateiktoje metodikoje. Tai yra labai įdomu, nes tikslumas tokiu atveju didesnis kai yra 190 km. ir siekia net virš 99 %. O mažiausias tikslumas yra kai gedimo atstumas yra 70 km. ir siekia tik 90 – 97 % priklausomai nuo bandymo metodo ( be DC filtro 90 proc. su DC filtru 97 proc.).
6. Bandymų metu nustatyta, kad ši platforma veikia tiksliausiai kai gedimas yra tarp fazės – žemės. Tada tikslumas gali siekti iki 99 % priklausomai nuo pasirinktos metodikos atstumui iki gedimo vietos surasti. Taip pat bandymų metu nustatyta, kad šis metodas gali surasti mažiausias vidutines varžas ir jas atvaizduoti grafikuose kai gedimai yra 2 fazės arba 2 fazės – žemė. Tačiau platforma pati neišskaičiuoja gedimo vietos atstumo, tai reikia padaryti rankiniu būdu. Su realiais duomenimis ištrta, kad vienfazio gedimo metu kai nežinoma gedimo varža, šio algoritmo tikslumas yra 91,776 %. Skaičiavimų greitis viršija 80 sekundžių. O paklaida viršija 8 % priklausomai nuo gedimo varžos įtakos skaičiuojamiems duomenims.

7. Šis modelis lengvai gali būti pritaikytas su realiais duomenimis, tačiau atliekant skaičiavimus reikia įvertinti gedimo varžos įtaką. Norint kuo tiksliau apskaičiuoti gedimo vietą.

Ši platforma puikiai skaičiuoja gedimo vietą vienos fazės avarijų metu. Ji visiškai netinkama naudoti kai gedimai yra tarp tirjų fazių ar trijų fazių ir žemės. Galima naudoti su realiais duomenimis kai gedimai yra tarp dviejų fazių ir dviejų fazių ir žemės.

8. Sukurta platforma yra ekonomiška, pigi bei greitai pritaikoma su realiais duomenimis elektros energetikos sistemoje.

## LITERATŪROS SĄRAŠAS

1. AŽUBALIS, V. KORYZNA, J. Trumpieji jungimai elektros sistemose. Vilnius: Raidės, 1986, p. 99.
2. SAHA, M. ROSOŁOWSKI, E. IZYKOWSKI, J.J. Fault Location on Power Networks 2010, p. 425.:
3. STRZELECKI, R. BENYSEK, G. Power Electronics in Smart Electrical Energy Networks. Poland. Springer. 2008, p. 414.
4. MASIOKAS, S. Elektrotechnika. Kaunas. Candela. 1994, p.430.
5. SANT, M.T, PAITHANKAR, Y.G. Online digital fault locator for overhead transmission line. [Interaktyvus] 1979. IEE Proc C, vol. 126(11), p.1181–1185. [Žiūrėta 2017m. lapkričio mėn]. Prieiga per internetą: <https://ieeexplore.ieee.org/document/5253443/>
6. „MATLAB“ user's guide. The Math Works Inc., Natick. 2013.
7. „MATLAB“ pradžiamokslis TECHNOLOGIJOS. [Interaktyvus] [Žiūrėta 2017 lapkričio mėn] Prieiga per internetą: [http://www.technologijos.lt/n/technologijos/it/straipsnis/“MATLAB“\\_pradziamokslis?name=straipsnis-2653](http://www.technologijos.lt/n/technologijos/it/straipsnis/“MATLAB“_pradziamokslis?name=straipsnis-2653)
8. PHADKE, A.G. THORP, J.S. Computer Relaying for Power Systems.[Interaktyvus] Research Studies Press Ltd., John Wiley & Sons, Inc., New York. 1988, p. 344. [Žiūrėta 2018 sausio mėn] Prieiga per internetą: [https://www.researchgate.net/publication/44551087\\_Computer\\_relaying\\_for\\_power\\_systems\\_Arun\\_G\\_Phadke\\_James\\_S\\_Thorpe](https://www.researchgate.net/publication/44551087_Computer_relaying_for_power_systems_Arun_G_Phadke_James_S_Thorpe)
9. SACHDEV, M.S. Advancement in microprocessor based protection and communication.[Interaktyvus] IEEE Tutorial Course, IEEE PES, IEEE Catalog Number: 97TP120-0. 1997, p. 91. [Žiūrėta 2018 sausio mėn] Prieiga per internetą: <http://www.pes-psrc.org/kb/published/reports/UNTAR-Ed2.pdf>
10. SUONAN, J.L. SONG, G.B. ZHANG, J.K. A novel short-window algorithm of phasor computation for EHV line protection.[Interaktyvus] Electric Power System Res.2005, vol.73(1), p. 1–7. [Žiūrėta 2018 sausio mėn] Prieiga per internetą: <http://opac.vimaru.edu.vn/edata/EJournal/2005/Electric%20power%20systems%20research/v73su1.1.pdf>
11. JOHNS, A.T. SALMAN, S.K. Digital Protection for Power Systems.[Interaktyvus] IEEE Power Series 15, Peter Peregrinus Ltd., London. 1997 vol. 28(2) p. 216. [Žiūrėta 2017 gruodžio mėn] Prieiga per internetą:

[https://www.researchgate.net/publication/233530442\\_Digital\\_Protection\\_for\\_Power\\_Systems?iepl%5BgeneralViewId%5D=X0mSfILGV1InvezUHUTK90YofqmAnI09ptrO&iepl%5Bcontexts%5D%5B0%5D=searchReact&iepl%5BviewId%5D=JRwJjSIPOamKYX0QRPwMSgZkHY1mgGshGzL3&iepl%5BsearchType%5D=publication&iepl%5Bdata%5D%5BcountLessEqual20%5D=1&iepl%5Bdata%5D%5BinteractedWithPosition1%5D=1&iepl%5Bdata%5D%5BwithoutEnrichment%5D=1&iepl%5Bposition%5D=1&iepl%5BrgKey%5D=PB%3A233530442&iepl%5BtargetEntityId%5D=PB%3A233530442&iepl%5BinteractionType%5D=publicationTitle](https://www.researchgate.net/publication/233530442_Digital_Protection_for_Power_Systems?iepl%5BgeneralViewId%5D=X0mSfILGV1InvezUHUTK90YofqmAnI09ptrO&iepl%5Bcontexts%5D%5B0%5D=searchReact&iepl%5BviewId%5D=JRwJjSIPOamKYX0QRPwMSgZkHY1mgGshGzL3&iepl%5BsearchType%5D=publication&iepl%5Bdata%5D%5BcountLessEqual20%5D=1&iepl%5Bdata%5D%5BinteractedWithPosition1%5D=1&iepl%5Bdata%5D%5BwithoutEnrichment%5D=1&iepl%5Bposition%5D=1&iepl%5BrgKey%5D=PB%3A233530442&iepl%5BtargetEntityId%5D=PB%3A233530442&iepl%5BinteractionType%5D=publicationTitle)

12. BENMOUYAL, G. Removal of DC-offset in current waveforms using digital mimic filtering.[Interaktyvus] IEEE Trans Power Deliv. 1995, vol. 10(2), p. 621–630. [Žiūrėta 2017 gruodžio. mėn] Prieiga per internetą:

[https://www.researchgate.net/publication/3272743\\_Removal\\_Of\\_Dc-Offset\\_In9\\_Current\\_Waveforms\\_Using\\_Digital\\_Mimic\\_Filtering?iepl%5BgeneralViewId%5D=50i6ktL5MfaIVsualVivVk9q0MIMvK8WV6OZ&iepl%5Bcontexts%5D%5B0%5D=searchReact&iepl%5BviewId%5D=idhwoXvOltSnAuCrhlhskoVDO2D0vNhimhxX&iepl%5BsearchType%5D=publication&iepl%5Bdata%5D%5BcountLessEqual20%5D=1&iepl%5Bdata%5D%5BinteractedWithPosition1%5D=1&iepl%5Bdata%5D%5BwithoutEnrichment%5D=1&iepl%5Bposition%5D=1&iepl%5BrgKey%5D=PB%3A3272743&iepl%5BtargetEntityId%5D=PB%3A3272743&iepl%5BinteractionType%5D=publicationTitle](https://www.researchgate.net/publication/3272743_Removal_Of_Dc-Offset_In9_Current_Waveforms_Using_Digital_Mimic_Filtering?iepl%5BgeneralViewId%5D=50i6ktL5MfaIVsualVivVk9q0MIMvK8WV6OZ&iepl%5Bcontexts%5D%5B0%5D=searchReact&iepl%5BviewId%5D=idhwoXvOltSnAuCrhlhskoVDO2D0vNhimhxX&iepl%5BsearchType%5D=publication&iepl%5Bdata%5D%5BcountLessEqual20%5D=1&iepl%5Bdata%5D%5BinteractedWithPosition1%5D=1&iepl%5Bdata%5D%5BwithoutEnrichment%5D=1&iepl%5Bposition%5D=1&iepl%5BrgKey%5D=PB%3A3272743&iepl%5BtargetEntityId%5D=PB%3A3272743&iepl%5BinteractionType%5D=publicationTitle)

13. ISAKSSON, I. Digital protective relaying through recursive least squares identification.[Interaktyvus] IEE Proc – Gener Transm Distrib. 1988, vol. 135(5), p. 441–449. [Žiūrėta 2018 balandžio. mėn] Prieiga per internetą:

[https://www.researchgate.net/publication/3360279\\_Digital\\_protective\\_relaying\\_through\\_recursive\\_least-squares\\_identification?iepl%5BgeneralViewId%5D=K2or1gOkZ05af38NGCNO1ZxFsVJeYMFV uejF&iepl%5Bcontexts%5D%5B0%5D=searchReact&iepl%5BviewId%5D=The2KvHmgDIEPfyTJuApHGoB4dchi3oxEraJ&iepl%5BsearchType%5D=publication&iepl%5Bdata%5D%5BcountLessEqual20%5D=1&iepl%5Bdata%5D%5BinteractedWithPosition1%5D=1&iepl%5Bdata%5D%5BwithoutEnrichment%5D=1&iepl%5Bposition%5D=1&iepl%5BrgKey%5D=PB%3A3360279&iepl%5BtargetEntityId%5D=PB%3A3360279&iepl%5BinteractionType%5D=publicationTitle](https://www.researchgate.net/publication/3360279_Digital_protective_relaying_through_recursive_least-squares_identification?iepl%5BgeneralViewId%5D=K2or1gOkZ05af38NGCNO1ZxFsVJeYMFV uejF&iepl%5Bcontexts%5D%5B0%5D=searchReact&iepl%5BviewId%5D=The2KvHmgDIEPfyTJuApHGoB4dchi3oxEraJ&iepl%5BsearchType%5D=publication&iepl%5Bdata%5D%5BcountLessEqual20%5D=1&iepl%5Bdata%5D%5BinteractedWithPosition1%5D=1&iepl%5Bdata%5D%5BwithoutEnrichment%5D=1&iepl%5Bposition%5D=1&iepl%5BrgKey%5D=PB%3A3360279&iepl%5BtargetEntityId%5D=PB%3A3360279&iepl%5BinteractionType%5D=publicationTitle)

14. SACHDEV, M. NAGPAL, M. A recursive least error squares algorithm for power system relaying and measurement application.[Interaktyvus] IEEE Trans on Power Deliv. 1991, vol. 6(3), p.1008–1015. [Žiūrėta 2018 vasario. mėn] Prieiga per internetą:

<https://ieeexplore.ieee.org/document/85841/>

15. ROSOŁOWSKI, E. MICHALIK, M. Fast identification of symmetrical components by use of a state observer.[Interaktyvus] IEE Proc – Gener Transm Distrib. 1994, vol. 141(6), p. 617–622. [Žiūrėta 2018 balandžio. mėn] Prieiga per internetą: [https://www.researchgate.net/publication/3354085\\_Fast\\_identification\\_of\\_symmetrical\\_components\\_by\\_use\\_of\\_a\\_state\\_observer?iepl%5BgeneralViewId%5D=y81JSCTdVy5KZbjDnAattLg7hlTNQqWeXk0m&iepl%5Bcontexts%5D%5B0%5D=searchReact&iepl%5BviewId%5D=0yYJy1XIQw6jK791z5mQI4iEv1q2em4Aws7J&iepl%5BsearchType%5D=publication&iepl%5Bdata%5D%5BcountLessEqual20%5D=1&iepl%5Bdata%5D%5BinteractedWithPosition1%5D=1&iepl%5Bdata%5D%5BwithoutEnrichment%5D=1&iepl%5Bposition%5D=1&iepl%5BrgKey%5D=PB%3A3354085&iepl%5BtargetEntityId%5D=PB%3A3354085&iepl%5BinteractionType%5D=publicationTitle](https://www.researchgate.net/publication/3354085_Fast_identification_of_symmetrical_components_by_use_of_a_state_observer?iepl%5BgeneralViewId%5D=y81JSCTdVy5KZbjDnAattLg7hlTNQqWeXk0m&iepl%5Bcontexts%5D%5B0%5D=searchReact&iepl%5BviewId%5D=0yYJy1XIQw6jK791z5mQI4iEv1q2em4Aws7J&iepl%5BsearchType%5D=publication&iepl%5Bdata%5D%5BcountLessEqual20%5D=1&iepl%5Bdata%5D%5BinteractedWithPosition1%5D=1&iepl%5Bdata%5D%5BwithoutEnrichment%5D=1&iepl%5Bposition%5D=1&iepl%5BrgKey%5D=PB%3A3354085&iepl%5BtargetEntityId%5D=PB%3A3354085&iepl%5BinteractionType%5D=publicationTitle)
16. GIRGIS, A.A. BROWN, R.G. Application of Kalman filtering in computer relaying. [Interaktyvus] IEEE Trans on PAS. 1981, vol. 100(7), p. 3387–3397. [Žiūrėta 2018 vasario. mėn] Prieiga per internetą: <https://ieeexplore.ieee.org/document/4111018/>
17. ROSOŁOWSKI, E. IZYKOWSKI, J. KASZTENNY, B. Adaptive measuring algorithm suppressing a decaying DC component for digital protective relays.[Interaktyvus] Electric Power System Res. 2001, vol. 60(2), p. 99–105. [Žiūrėta 2018 kovo. mėn] Prieiga per internetą: [https://www.researchgate.net/publication/263572449\\_Adaptive\\_measuring\\_algorithm\\_suppressing\\_a\\_decaying\\_DC\\_component\\_for\\_digital\\_protective\\_relays](https://www.researchgate.net/publication/263572449_Adaptive_measuring_algorithm_suppressing_a_decaying_DC_component_for_digital_protective_relays)
18. WIZNIEWSKI.A. Digital Signal Processing in Power Protection and Control. [Interaktyvus] Springer, London. 2011, p. 316. [Žiūrėta 2018 kovo. mėn] Prieiga per internetą: [https://archive.org/stream/Digital\\_Signal\\_Processing\\_in\\_Power\\_System\\_Protection\\_and\\_Control/Digital\\_Signal\\_Processing\\_in\\_Power\\_System\\_Protection\\_and\\_Control\\_djvu.txt](https://archive.org/stream/Digital_Signal_Processing_in_Power_System_Protection_and_Control/Digital_Signal_Processing_in_Power_System_Protection_and_Control_djvu.txt)
19. ZAVECKAS, V. Elektrotechnikos pagrindai. Vilnius . Technika. 2012, p.91.
20. C37.111\_WG – POWER SYSTEM RELAYING COMMITTEE. Common Format for Transient Data Exchange (COMTRADE) for Power Systems.[Interaktyvus] IEEE Standard C37.111-2013. [Žiūrėta 2017m. rugpjūčio mėn]. Prieiga per internetą: <https://ieeexplore.ieee.org/document/6654239/>



## PRIEDAI

1 priedas

```
clear all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% COMTRADE FAILAS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%ComtradeFile='as2';%Vienfazis B fazė
%ComtradeFile='Sysa_1';%Vienfazis trumpas jungimas
%ComtradeFile='JoniskisB2';%Dvifazis trumpas jungimas
%ComtradeFile='REC';
%ComtradeFile='aleksotas';
%ComtradeFile='TelšiaiA2';
%ComtradeFile='telšiai3';
%ComtradeFile='Avarija_2';
%ComtradeFile='aleksotas L-KHE trip';
%ComtradeFile='kaunoE L-KHE1trip';

ComtradeFile='AleksotL-PE 09 April';

%ComtradeFile='PE L-Aleksot 09 April';
%ComtradeFile='Alytus 1 matavimai';
%ComtradeFile='newconfigM';
%ComtradeFile='L-Prienai_2017_09_22';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% nsig, analoginių ir diskretinių signalų ilgiai
% kodas tekstinio failo turinys
i=1;
Cfgfailas=strcat(ComtradeFile, '.cfg');
fid=fopen(Cfgfailas);
while 1
tline = fgetl(fid); %ieško pirmos eilutės
if ~ischar(tline), break, end
kodas{i,1}=tline;
%disp(tline) %perskaito visas eilutes cfg. failo
i=i+1;
end
fclose(fid); %uždaro fid failą (t.y cfg. failas )
%užkrautų duomenų apdorojimas
A=kodas{2,1}; %analoginiai signalai (šiuo atveju 8, tai priklauso nuo pasirinktų duomenų kiek jų yra), ima užkrauto cfg. failo duomenis kadangi viskas viename 1 stulpelyje ima 2 eilutę kur surašyta analoginiai, diskretiniai signalu kiekiai
kb=findstr(A, ','); %suranda jo vietą, ieško kablelio, kb reiškia kablelis. išrenka kablelius iš 2 eilutės 1 stulpelio t.y kad netrukdytų atrinkti kiek analoginių kiek diskretinių signalų kiekio
sD=findstr(A, 'D'); %suranda jo vietą faile. ieško diskretinio signalo, sD reiškia skaičius diskretinių signalų, t.y iš 2 eilutės ir 1 stulpelio ima ir ieško D raidės prieš ją parašytas skaičius reiškia signalo kiekį

%signalų kiekiai
```

```

nsig(1)=str2num(A(1:kb(1)));          %(visas signalų kiekis) nuo 1 skaičiaus iki pirmo kablelio jų
vieta, viso kiek signalų, skaičiuoja, nusako*** pasako kad duomenyse yra tiek signalų
nsig(2)=str2num(A(kb(1)+1:kb(2)-2));          %(visas kiekis signalų + kiek analoginių) po pirmo
kablelio skaičius iki antro kablelio -2, analoginių kiek signalų, skaičiuoja, nusako***
nsig(3)=str2num(A(kb(2)+1:sD-1));          %(visas kiekis signalų + kiek analoginių + kiek diskretinių
signalų) nuo 2 kablelio +1 iki D raidės -1 jų vieta. diskretinių kiek signalų, skaičiuoja, nusako***

%Analoginiai signalai
for i=1:nsig(2); %ima visus analoginius signalus kiek jų yra nuo 1 iki ... (pvz nuo 1 iki 9)
Akod=kodas{i+2,1}; %sudarinėja Akod, kuris reiškia toji eilutė viso cfg +2, t.y. nuo 3 eilutės
pradedame ir einame žemyn per visas eilutes, pirmo stulpelio, kodo faile

kb=findstr(Akod,','); % ima akod visas eilutes ir ieško kablelio, kablelius kiek yra skaičiuoja
ei=length(Akod); %eilutės ilgis (išrenka iš visos eilutės visus duomenis ir juos suskaičiuoja kiek
jų yra)
kbi=length(kb); % kablelių skaičius (išrenka iš visos eilutės kiek yra kablelių ir suskaičiuoja
kiek jų yra)
skA(i,1)=str2num(Akod(1:kb(1)-1)); %suteikia numerius str iš tekstinio pakeičia tekstą (i toji eilutė
pirmo stulpelio)
for j=2:5; % 2 3 4 5 ima
    n=3+j; % 5 6 7 8
    if (kb(n)+1)-(kb(n+1)-1)==1;skA(i,j)=0; %kur kalbėlis ten 0 turėtų būti
        else skA(i,j)=str2num(Akod(kb(n)+1:kb(n+1)-1));
    end
end
if kbi==9;skA(i,6)=str2num(Akod(kb(9)+1:(ei)));end
if kbi>9;skA(i,6)=str2num(Akod(kb(9)+1:kb(10)-1));end
for j=2:5; % 2 3 4 5 ima
    if kb(j)-kb(j-1)>0;numA{i,j-1}=Akod(kb(j-1)+1:kb(j)-1);
        else numA{i,j-1}=' ';
    end %surenka analoginius signalus tarp kablelių
end
end
[a1,a2]=size(numA);%a1-užrašų eilučių kiekis
%Diskretiniai signalai
for i=1:nsig(3)
    Dkod=kodas{i+2+nsig(2),1}; %skaičius diskretinius signalus ima i tąjį diskretinį signalą t.y šiuo
atveju
    kb=findstr(Dkod,','); %ieško kablelių toje diskretinėje eiluteje pasirinktoje
    kbi=length(kb); %skaičiuoja skaičių kablelius toje diskretinėje eiluteje
    ei=length(Dkod); %skaičiuoja eilutės ilgį
    numD{1,i}=Dkod(kb(1)+1:kb(2)-1);
    if ei>kb(kbi);skD(i)=str2num(Dkod(kb(kbi)+1));
        else skD(i)=0;end %išrenka diskretinius signalus
end
Dz=str2num(kodas{3+nsig(1),1});
time=str2num(kodas{5+nsig(1),1});
%Konfiguravimo duomenų pabaiga
AoB=kodas{8+nsig(1)}; %
Failas{1,1}='ComtradeFile';Failas{1,2}=ComtradeFile;Failas{1,3}=AoB;Failas{2,1}='Daznis';
%išrenka kuris yra dažnis configo faile
Failas{2,2}=kodas{3+nsig(1),1};Failas{2,3}='Hz'; % jam priskiria hz

```

```

if strcmp(AoB,'ASCII')==0; warning('Apdorojami BINARINIAI duomenis');end
Failas
Laikas=kodas{6+nsig(1),1} %priskiria kad tai laikas
numA
%Duomenų formavimas
Duomfailas=strcat(ComtradeFile,'.dat'); %užkrauna data failą
M = dlmread(Duomfailas,','); %atspausdina
%Rezultatų spausdinimas
[ani,n]=size(numA); %visus atrenka analoginius signalus ir spausdina
r{1,1}=numA{1,4};m=1;
re=r;lr=r;g=2;
for i=2:ani;
    re{i,1}=numA{i,4};
    if strcmp(numA{i,4},r)==0;
        r{g,1}=numA{i,4};g=g+1;
        m=m+1;
    end % ir sudeda i dua matricą
end
% "re" - dimensijų matrica stulpelis
% "r" - skirtingų dimensijų matrica stulpelis
rr=strcat(re{:}); % "rr" - visos dimensijos surašytos į eilutę be tarpų
dan=M(:,1:ani+2); % "dan" - laikas ir analoginiai duomenys iš *.dat failo.
sr=time(1)*dan(2,2);
if sr==0;sr=1000000; end % čia apskaičiuojamas laiko mastelis
g=1;Duom(:,1)=(dan(:,2))/sr; % suformuojamas laiko stulpelis
% m - parodo, kiek yra skirtingų dimensijų
for i=1:ani %ima visus analoginius
    Duom(:,i+1)=(M(:,i+2)*skA(i,2))+skA(i,3);
end
for i=1:m
    clear le duo;
    figure(1)
    subplot(m,1,i) % suformuojamas grafiku išdėstymas: 1 stulpelis ir m eilučių
    k=0;
    for j=1:ani
        if strcmp(numA{j,4},r(i,1))==1;
            k= k+1;
            duo(:,k)=Duom(:,j);
            le{k}=numA{j,1};
        end
    end
    plot(Duom(:,1),duo)
    xlabel('t laikas, s') % x-ašis
    ylabel(r{i,1}) % y-ašis
    legend(le) % legenda grafiko
    str=strcat(kodas{1,1},('r{i,1},'));
    title(str)
end
end

```

```

%pasitikrinti žingsnį ar 46, 47, 50 ar 100
clear all
load('DuomPradiniai.mat') %užkrauna pradinis duomenis
load('DuomXsXc.mat') % duomenis nufiltruotus
%Duomenų apdorojimas*****
T=Duom(:,1);
%T=DuomXsXc(:,1);
%UA=Duom(:,6); UB=Duom(:,5);UC=Duom(:,7);IA=Duom(:,2);IB=Duom(:,3);IC=Duom(:,4);
%UA=Duom(:,2); UB=Duom(:,3);UC=Duom(:,4);IA=Duom(:,5);IB=Duom(:,6);IC=Duom(:,7);
%UA=Duom(:,6); UB=Duom(:,7);UC=Duom(:,8);IA=Duom(:,2);IB=Duom(:,3);IC=Duom(:,4);
%UA=Duom(:,2); UB=Duom(:,4);UC=Duom(:,6);IA=Duom(:,3);IB=Duom(:,5);IC=Duom(:,7);
%UA=Duom(:,2); UB=Duom(:,3);UC=Duom(:,4);IA=Duom(:,5);IB=Duom(:,6);IC=Duom(:,7);
%pradiniai duomenys *****
UA=Duom(:,6); UB=Duom(:,7);UC=Duom(:,8);IA=Duom(:,2);IB=Duom(:,3);IC=Duom(:,4);
%
UA=DuomXSXC(:,6);UB=DuomXSXC(:,7);UC=DuomXSXC(:,8);IA=DuomXSXC(:,2);IB=Duom
XSXC(:,3);IC=DuomXSXC(:,4);
%nufiltruoti duomenys*****viršui
%UA=Duom(:,7); UB=Duom(:,8);UC=Duom(:,9);IA=Duom(:,2);IB=Duom(:,3);IC=Duom(:,4);
%Bendroji dalis
clear UAef UBef UCef IAef IBef ICef INef BP GP Bp Gp Fua Fub Fuc Fia Fib Fic
clear UAes UBes UCes Imm Immt Uvid Ivid Uef Ief
clear UAvs UBvs UCvs IAes IBes ICes INes IAvs IBvs ICvs INvs
clear B G Bvid Gvid Bs Gs Yvid Yk Zs
%
Iosk=sum([IA';IB';IC']);IN=-1*Iosk'; %nulinės sekos*****
%Signalų keitimas į efektingas vertes
tk=T(2,1); %laikas t *****
Del=2*pi*50*tk; % ?%diskretizacijos dažnis (koks laiko tarpas tarp jų )
m=length(T);
for k=1:m-2;
    xua=UA(k:k+2);xub=UB(k:k+2);xuc=UC(k:k+2);
    xia=IA(k:k+2); xib=IB(k:k+2); xic=IC(k:k+2); xin=IN(k:k+2);
    [aua(k),bua(k)]=Fakre(xua,Del);%PF (naudajama papildoma funkcija) %funkcija išskiria
    akyviasias reaktyviasias dalis
    [aub(k),bub(k)]=Fakre(xub,Del); %PF naudojamos įtampos pastumtos k
    [auc(k),buc(k)]=Fakre(xuc,Del); %PF skaičiuoja dedamąsias iš sinusoides
    %srovių dedamosios
    [aia(k),bia(k)]=Fakre(xia,Del); %PF srovės dedamąsias
    [aib(k),bib(k)]=Fakre(xib,Del); %PF
    [aic(k),bic(k)]=Fakre(xic,Del); %PF
    [ain(k),bin(k)]=Fakre(xin,Del); %PF
    %Reaktyvas
    Ba(k)=((aua(k)*bia(k))-(bua(k)*aia(k)))/(aua(k)^2+bua(k)^2); % pakaičiuojame laidumus iš
    srovių ir itampų dedamųjų
    Bb(k)=((aub(k)*bib(k))-(bub(k)*aib(k)))/(aub(k)^2+bub(k)^2);
    Bc(k)=((auc(k)*bic(k))-(buc(k)*aic(k)))/(auc(k)^2+buc(k)^2);
    B=[Ba;Bb;Bc];B=B';
    %Aktyvas

```

```

Ga(k)=((aua(k)*aia(k))+(bua(k)*bia(k)))/(aua(k)^2+bua(k)^2);
Gb(k)=((aub(k)*aib(k))+(bub(k)*bib(k)))/(aub(k)^2+bub(k)^2);
Gc(k)=((auc(k)*aic(k))+(buc(k)*bic(k)))/(auc(k)^2+buc(k)^2);
G=[Ga;Gb;Gc];G=G';
end,
%Efektinių verčių signalai
UAef=sqrt((aua.^2+bua.^2)/2); % iš momentinių paskaičiuoja efektines
UBef=sqrt((aub.^2+bub.^2)/2);
UCef=sqrt((auc.^2+buc.^2)/2);
IAef=sqrt((aia.^2+bia.^2)/2);
IBef=sqrt((aib.^2+bib.^2)/2);
ICef=sqrt((aic.^2+bic.^2)/2);
INef=sqrt((ain.^2+bin.^2)/2);
Uef=[UAef;UBef;UCef];Uef=Uef';
Ief=[IAef;IBef;ICef;INef];Ief=Ief';
m=m-2;
%Efektinių verčių vidurkinimo žingsnis 100 pagal dest (pagal dc comp gal 46,47,50)
%Laidumo vidurkinimas žingsniu 100 (pagal dc cont šalinimą gal 47 žingsniai patikrinti)
vz=47; %atskirti laidumų vidurkinimų žingsnius
for k=1:m-vz;
    Uvid(k,:)=mean(Uef(k:k+vz,:)); %kas tam tikrą žingsni skaičiuoja vidurkius iš 100 ir taškas kad
    greičiau skaičiuotų
    Ivid(k,:)=mean(Ief(k:k+vz,:));
    Bvids(k,:)=mean(B(k:k+vz,:));
    Gvids(k,:)=mean(G(k:k+vz,:));
    mvid=length(Uvid);
end %kada išjungia jungtuvą varža begalybė todėl ja pašalina
Yvid=Bvids.*Gvids./(sqrt(Bvids.^2+Gvids.^2));%pilnasis vidutinis laidumas
Bvid=abs(Bvids);Gvid=abs(Gvids);
% Skaičiavimai
%
Fkaml=atand(Bvids./Gvids);
%Efektinių signalų išvestinės
UAes=UAef(2:end)-UAef(1:end-1);UAes(2:end+1)=UAes(1:end);
UBes=UBef(2:end)-UBef(1:end-1);UBes(2:end+1)=UBes(1:end);
UCes=UCef(2:end)-UCef(1:end-1);UCes(2:end+1)=UCes(1:end);
IAes=IAef(2:end)-IAef(1:end-1);IAes(2:end+1)=IAes(1:end);
IBes=IBef(2:end)-IBef(1:end-1);IBes(2:end+1)=IBes(1:end);
ICes=ICef(2:end)-ICef(1:end-1);ICes(2:end+1)=ICes(1:end);
INes=INef(2:end)-INef(1:end-1);INes(2:end+1)=INes(1:end);

%Vidutinių reikšmių signalų išvestinės
UAvs=Uvid((2:end),1)-Uvid((1:end-1),1);UAvs(2:end+1)=UAvs(1:end);
UBvs=Uvid((2:end),2)-Uvid((1:end-1),2);UBvs(2:end+1)=UBvs(1:end);
UCvs=Uvid((2:end),3)-Uvid((1:end-1),3);UCvs(2:end+1)=UCvs(1:end);
IAvs=Ivid((2:end),1)-Ivid((1:end-1),1);IAvs(2:end+1)=IAvs(1:end);
IBvs=Ivid((2:end),2)-Ivid((1:end-1),2);IBvs(2:end+1)=IBvs(1:end);
ICvs=Ivid((2:end),3)-Ivid((1:end-1),3);ICvs(2:end+1)=ICvs(1:end);
INvs=Ivid((2:end),4)-Ivid((1:end-1),4);INvs(2:end+1)=INvs(1:end);
%Vidutinės varžos
Bmax=max(max(B))*2;Gmax=max(max(G))*2;Ymax=sqrt(Bmax.^2+Gmax.^2)*2;%Didžiausia
laidumo vertė

```

```

Bmin=(abs(min(min(B(:,1)))))/2;Gmin=(abs(min(min(G(:,1)))))/2;Ymin=sqrt(Bmin.^2+Gmin.^2)/2
;%Mažiausia laidumo vertė
%
m1=length(Bvid);
Bk=Bvid;Gk=Gvid;Yk=Yvid;
db1=find(Bvid(:,1)<=Bmin);dg1=find(Gvid(:,1)<=Gmin);dy1=find(Yvid(:,1)<=Ymin);
Bk(db1,1)=Bmin;Gk(dg1,1)=Gmin;Yk(dy1,1)=Ymin;
db2=find(Bvid(:,2)<=Bmin);dg2=find(Gvid(:,2)<=Gmin);dy2=find(Yvid(:,2)<=Ymin);
Bk(db2,2)=Bmin;Gk(dg2,2)=Gmin;Yk(dy2,2)=Ymin;
db3=find(Bvid(:,3)<=Bmin);dg3=find(Gvid(:,3)<=Gmin);dy3=find(Yvid(:,3)<=Ymin);
Bk(db3,3)=Bmin;Gk(dg3,3)=Gmin;Yk(dy3,3)=Ymin;
%
%Varžų skaičiavimas
Xs(:,1)=Bk(:,1).\1;Xs(:,2)=Bk(:,2).\1;Xs(:,3)=Bk(:,3).\1;% Skaičiuojama reaktyvioji varža
Xs=Xs/2;
Rs(:,1)=Gk(:,1).\1;Rs(:,2)=Gk(:,2).\1;Rs(:,3)=Gk(:,3).\1;% Skaičiuojama aktyvioji varža
Rs=Rs/2;
Zs(:,1)=Yk(:,1).\1;Zs(:,2)=Yk(:,2).\1;Zs(:,3)=Yk(:,3).\1;% Skaičiuojama pilnutinė varža
Zs=Zs/2;
%Kampo skaičiavimas
Fkamv=atand(Xs./Rs);
%
%Avarijos analizė
%Signalų fazės
Fua=atand(bua./aua); %skaičiuoja kaip fazė kinta avarijos metu
Fub=atand(bub./aub);
Fuc=atand(buc./auc);
Fia=atand(bia./aia);
Fib=atand(bib./aib);
Fic=atand(bic./aic);
Fin=atand(bin./ain);
m2=length(Fua);
FU=[Fua' Fub' Fuc'];
FI=[Fia' Fib' Fic'];
% varžos išvestinės skaičiavimas
iZ=isv(Zs,1); %PF
iX=isv(Xs,1); %PF
iR=isv(Rs,1); %PF
%Avarijos laikas
[a1,b1]=max(iZ);%varžos išvestinės maksimumo radimas
[a2,b2]=max(max(iZ)); %fazės išrinkimas
c=b1(b2);%fazės nustatymas
[a3,b3]=min(iZ(1:c,b2));
Tavarijos=[T(b3) T(b1(b2))];%Avarijos laiko [pradžia pabaiga]
dTavarijos=Tavarijos(2)-Tavarijos(1);%Avarijos trukmė sekundėmis
%AKI pauzė
ii=find(iZ(c:end,3)==0)+c;
Taki=[T(ii(1)) T(ii(end))];% AKI pauzės laiko [ pradžia pabaiga]
dTaki=Taki(2)-Taki(1); %AKI pauzės laiko tarpas
Tavarijos
dTavarijos
Taki

```

```

dTaki
%GRAFIKAI
figure(1);%Kiekvienos fazės įtampa hold on kad vaizduotu tame pačiame lange
hold on;
subplot(3,1,1);plot(T(1:m),[UA(1:m) UAef]);xlabel('Fazes A itampa');grid on
subplot(3,1,2);plot(T(1:m),[UB(1:m) UBef]);xlabel('Fazes B itampa');grid on
subplot(3,1,3);plot(T(1:m),[UC(1:m) UCef]);xlabel('Fazes C itampa');grid on
figure(2);%Kiekvienos fazės srovė

hold on;
subplot(4,1,1);plot(T(1:m),[IA(1:m) IAef]);xlabel('Fazes A srove');grid on
subplot(4,1,2);plot(T(1:m),[IB(1:m) IBef]);xlabel('Fazes B srove');grid on
subplot(4,1,3);plot(T(1:m),[IC(1:m) ICef]);xlabel('Fazes C srove');grid on
subplot(4,1,4);plot(T(1:m),[IN(1:m) INef]);xlabel('3Io srove');grid on
figure(3);%Vidutinė reaktyvioji varža

hold on;
plot(T(1:m1),Xs);xlabel('Vidutine reakyvioji varza X');grid on
figure(4);%Vidutinė aktyvioji varža

hold on;
plot(T(1:m1),Rs);xlabel('Vidutine akyvioji varza R');grid on
figure(5);%Z varžos kampas

hold on;
plot(T(1:m1),Fkamv);xlabel('Varzos faziu kampas');grid on
%%
figure(6);%Vidutinė pilnutinė varža
hold on;
plot(T(1:m1),Zs);xlabel('Vidutine pilnoji varza Z');grid on
[minRV, minR] = min(Zs,[],1);
[minCV, minC] = min(minRV,[],2);
ZZsk = minRV(minC);
plot(T(minR(minC)),ZZsk,'ro');
figure(4); hold on;
akyviojiVarza = Rs(minR(minC));
[minRVRs, minRRs] = min(Rs,[],1);
[minCVRs, minCRs] = min(minRVRs,[],2);
RRsk = minRVRs(minCRs);
plot(T(minRRs(minCRs)),RRsk,'bo'); %RRsk
Zsk = 20.75;
kint= (ZZsk) / Zsk;
kint
%%
figure(7)

hold on;
plot(T(1:m1),iZ);xlabel('Z isvestine');grid on
%figure(8)
%plot(T(1:m2),FI);xlabel('Sroviu fazes');grid on
%%
kampas = 64; %pasirenkamas kampas

```

```

ISMkampas = Fkamv>kampas;
ISLkampas = Fkamv<=kampas;
ISMBound = ISMkampas(1:end-1,:) & ISLkampas(2:end,:); % randa, kur kampas kerta nustatytą
vertę
IN = find(ISMBound == 1);
if ~isempty(IN) % jeigu yra tokia riba, kai kirto 64
    [I,J] = ind2sub(size(ISMBound),IN);
    [rowInd, minJ] = min(I);
    colInd = J(minJ);
    time = T(rowInd+1)-(T(rowInd+1)-T(rowInd))*...
        (Fkamv(rowInd+1,colInd)-kampas)/...
        (Fkamv(rowInd+1,colInd)-Fkamv(rowInd,colInd));
    figure(5); hold on;
    plot(time,kampas,'m*');
    Zsk = 20.75;
    ZZsk = Zs(rowInd+1,colInd) - (T(rowInd+1)-time)*...
        (Zs(rowInd+1,colInd)-Zs(rowInd,colInd))/...
        (T(rowInd+1)-T(rowInd));
    figure(6); hold on;
    plot(time,ZZsk,'m*');
    RRsk = Rs(rowInd+1,colInd) - (T(rowInd+1)-time)*...
        (Rs(rowInd+1,colInd)-Rs(rowInd,colInd))/...
        (T(rowInd+1)-T(rowInd));
    figure(4); hold on;
    plot(time,RRsk,'m*');
    kint= (ZZsk )/ Zsk; %RRsk

    kint
else
    display('reiksme nekerta kreiviu')
end

```

%Alytaus Seirijai linijos ilgis 46.51km linija 110kv



```

% funkcijos [Xc,Xs,XR,XI,del_c,del_s]=dc_lsm(N,f1,x_k);
% įėjimai N – skaičius of žingsnių per periodą
% f1 – dažnis sistemos
% x_k – įėjimo signalai užfiksuoti
% išėjimai: Xc, Xs – komponentai koreguoto fazoriaus
% XR, XI – komponentai fazoriaus (Fourier alg.)
% del_c, del_s – koreguoti signalai
clear all
close all
load('C:\Users\Arnas\Desktop\magistro ataskaita\bandimas\work\DuomPradiniai.mat')
N=47; %taškų suma periode (gal 50 gal 48, 47 gal 46 patikrinti)
f1=50;
a=2*pi/N;
DuomXSXC = zeros(size(Duom)); %sukuriame didelę nulinę matricą tuščią, ten krauname
duomenis, kad galėtume išsaugoti
for k = 2:8
    x_k=Duom(:,k);
    t1=1/f1; % sistemos dažnis periode
    tamx=0.3; % max. laikas nykimo
    tamn=0.004; % min. laikas nykimo
    rsmx=exp(-t1/(N*tamx));
    rsmn=exp(-t1/(N*tamn));
    for i=1:N, % koreliacijos funkcijos
        hR(i)=2*(cos((i-1)*a))/N;
        hI(i)=-2*(sin((i-1)*a))/N;
    end;
    ns=max(size(x_k)); % skaičius duomenų įėjimų
    j=1;
    % pagrindinis laiko ciklas
    for i=1:ns,
        dx=x_k(i);
        if i>N,
            dx=dx-x_k(i-N); % signalo skirtumas
        end;
        if i>1,
            XR(i)=XR(i-1)+hR(j)*dx; % pilno periodo fourrier
            XI(i)=XI(i-1)+hI(j)*dx; % rekursinis algoritmas
            Xa(i)=Xa(i-1)+dx; % dc dedamoji
            if abs(Xa(i-1))>0,
                rs(i)=Xa(i)/Xa(i-1);
                if (rs(i)>1)|(rs(i)<0),
                    rs(i)=1;
                elseif rs(i)<rsmn,
                    rs(i)=rsmn;
                end,
            else,
                rs(i)=1;
            end;
        else,
            rs(i)=1;
        end;
    end,
end,

```

```

XR(i)=hR(i)*dx; % pirmas etapas
XI(i)=hI(i)*dx;
Xa(i)=x_k(i);
rs(i)=1;
end;
p=(1-rs(i))/(1+rs(i)*(rs(i)-2*cos(a)));
pc=p*(cos(a)-rs(i));
ps=p*sin(a);
del_c(i)=(pc*hR(j)+ps*hI(j))*Xa(i); % korekcijos
del_s(i)=(ps*hR(j)-pc*hI(j))*Xa(i);
Xc(i)=XR(i)-del_c(i);
Xs(i)=XI(i)+del_s(i);
j=j+1;
if j>N,
    j=1;
end;

end; % *****bandymai k stulpelio skaičius ne mažiau nei 4
if k <= 4
figure(1); hold on;
plot(Duom(:,1),XR)

figure(2); hold on;
plot(Duom(:,1),XI);
figure(3); hold on;
plot(Duom(:,1),Xc);
figure(4); hold on;
plot(Duom(:,1),Xs);
figure(5); hold on;
plot(Duom(:,1),del_c);
figure(6); hold on;
plot(Duom(:,1),del_s);
figure(7); hold on;
plot(Duom(:,1),x_k);
plot(Duom(:,1),Xc'.*cos(Duom(:,1)*2*pi*f1)+Xs'.*sin(Duom(:,1)*2*pi*f1));
end
%atkuria nufiltruotą signalą xc cosinuso xs sinusas
DuomXSXC(:,k) = Xc'.*cos(Duom(:,1)*2*pi*f1)-Xs'.*sin(Duom(:,1)*2*pi*f1);
end
save('C:\Users\Arnas\Desktop\magistro ataskaita\bandimas\work\DuomXcXs.mat','DuomXSXC');
%save saugo duomenis nufiltruotus, viską ką reikia pasiliekam kitą išvalom
figure(1); hold on;
title('XR'); grid on;
legend('IA','IB','IC');
xlabel('time, s');
ylabel('A');
figure(2); hold on;
title('XI'); grid on;
legend('IA','IB','IC')
xlabel('time, s');
ylabel('A');
figure(3); hold on;

```

```
title('Xc'); grid on;
legend('IA','IB','IC')
xlabel('time, s');
ylabel('A');
figure(4); hold on;
title('Xs');grid on;
legend('IA','IB','IC')
figure(5); hold on;
title('del_c'); grid on;
legend('IA','IB','IC')
xlabel('time, s');
ylabel('A');
figure(6); hold on;
title('del_s'); grid on;
legend('IA','IB','IC')
xlabel('time, s');
ylabel('A');
```