



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

ELEKTRONIKOS INŽINERIJOS FAKULTETAS

TELEKOMUNIKACIJŲ INŽINERIJOS KATEDRA

Saulius Dreinius

**BEVIELIŲ RYŠIO TINKLŲ BAZINIŲ STOČIŲ SUKURIAMŲ
ELEKTROMAGNETINIŲ LAUKŲ INTENSYVUMO SUMAŽINIMO
GALIMYBIŲ TYRIMAS**

**ANALYSIS OF WIRELESS COMMUNICATION NETWORK BASE STATIONS
ELECTROMAGNETIC FIELD STRENGTH REDUCTION POSSIBILITIES**

Baigiamasis magistro darbas

Telekomunikacijų inžinerijos studijų programa, valstybinis kodas 621H64003

Telekomunikacijų technologijų specializacija

Elektronikos inžinerijos studijų kryptis

Vilnius, 2013

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

ELEKTRONIKOS INŽINERIJOS FAKULTETAS

TELEKOMUNIKACIJŲ INŽINERIJOS KATEDRA

TVIRTINU
Katedros vedėjas

(Parašas)

(Vardas, pavardė)

(Data)

Saulius Dreinius

**BEVIELIŲ RYŠIO TINKLŲ BAZINIŲ STOČIŲ SUKURIAMŲ
ELEKTROMAGNETINIŲ LAUKŲ INTENSYVUMO SUMAŽINIMO
GALIMYBIŲ TYRIMAS**

**ANALYSIS OF WIRELESS COMMUNICATION NETWORK BASE STATIONS
ELECTROMAGNETIC FIELD STRENGTH REDUCTION POSSIBILITIES**

Baigiamasis magistro darbas

Telekomunikacijų inžinerijos studijų programa, valstybinis kodas 621H64003

Telekomunikacijų technologijų specializacija

Elektronikos inžinerijos studijų kryptis

Vadovas

doc. dr. Ričardas Visvaldas Pocius

(Parašas)

(Data)

Lietuvių kalbos konsultantas

Vaida Buivydienė

(Parašas)

(Data)

Vilnius, 2013

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
TELEKOMUNIKACIJŲ INŽINERIJOS KATEDRA

TVIRTINU
Katedros vedėjas

Elektronikos inžinerijos studijų kryptis

Telekomunikacijų inžinerijos (621H64003) studijų programa

Telekomunikacijų technologijos specializacija

(Parašas)

Artūras Medeišis

(Vardas, pavardė)

(Data)

**BAIGIAMOJO MAGISTRO DARBO
UŽDUOTIS**

.....Nr.

Vilnius

Studentui Sauliui Dreiniui

Baigiamojo darbo tema: Bevielių ryšio tinklų bazinių stočių sukuriamų elektromagnetinių laukų intensyvumo sumažinimo galimybių tyrimas (*angl.* Analysis of wireless communication network base stations electromagnetic field strength reduction possibilities)

patvirtinta 201...m. d. dekanų potvarkiu Nr.

Baigiamojo darbo užbaigimo terminas 201...m. d.

BAIGIAMOJO DARBO UŽDUOTIS:

Darbo tikslas: Ištirti bevielių ryšio tinklų bazinių stočių sukuriamos elektromagnetinės spinduliuotės intensyvumo sumažinimo galimybes artimoje siųstuvų aplinkoje plėtojant egzistuojančius bevielio ryšio tinklus plėtros ir diegiant naujas technologijas.

Aprašo turinys:

1. Įvadas. Užduoties analizė.
2. EML vertinimo būdai.
3. EML vertinimo modelių realizacija.
4. EML įvertinimo programos kūrimas.
5. EML minimizavimo galimybių tyrimas.
6. Darbo rezultatų apibendrinimas.
7. Literatūros sąrašas.

Vadovas
(Parašas)

doc. dr. Ričardas Visvaldas Pocius
(Moksl. laipsnis/pedag.vardas, vardas, pavardė)

Užduotį gavau

(Parašas)

Saulius Dreinius

(Vardas, pavardė)

(Data)

Vilniaus Gedimino technikos universitetas
Elektronikos fakultetas
Telekomunikacijų inžinerijos katedra

ISBN ISSN
Egz. sk.
Data-.....-.....

Antrosios pakopos studijų **Telekomunikacijų inžinerijos** programos baigiamasis darbas 4

Pavadinimas **Bevielių ryšio tinklų bazinių stočių sukuriamų elektromagnetinių laukų intensyvumo sumažinimo galimybių tyrimas**

Autorius **Saulius Dreinius**

Vadovas **doc. dr. Ričardas Visvaldas Pocius**

Kalba: lietuvių

Anotacija

Šiame baigiamajame magistro darbe tiriama bevielių ryšio tinklų bazinių stočių sukuriamos elektromagnetinės spinduliuotės intensyvumo sumažinimo galimybė artimoje siųstuvų aplinkoje. Aptarti bazinių stočių, antenų parametrai, aprašytos galiojančios higienos normos. Išnagrinėta metodika, leidžianti apskaičiuoti antenos sukuriamą elektromagnetinio lauko stiprį. Aptariamas eksperimentinis elektromagnetinių laukų intensyvumo tyrimas ir jo rezultatai. Patiekama sukurta programa, kuri modeliuoja spinduliavimo intensyvumą artimoje zonoje, nustatytas draudžiamąsias zonas atvaizduoja žemėlapyje, geba automatiškai parinkti parametrus siekiant sumažinti spinduliavimo intensyvumą pasirinktuose taškuose. Atlikus modeliavimus, aptariami parametrai, kuriuos keičiant galima sumažinti EML energijos srauto tankį.

Darbą sudaro 7 skyriai. Darbo apimtis — 53 p. teksto be priedų, tekste yra 31 iliustracija, 21 bibliografinis šaltinis. Atskirai pridedami darbo priedai.

Prasminiai žodžiai: elektromagnetinės spinduliuotės intensyvumas, energijos srauto tankis, EML sumažinimas, įvertinimo programa, intensyvumo modeliavimas

Vilnius Gediminas Technical University
Faculty of Electronics
Department of Telecommunication Engineering

ISBN ISSN
Copies No.
Date-.....-.....

Master Degree Studies **Telecommunications Engineering** study programme Final Work 4

Title **Analysis of Wireless Communication Network Base Stations Electromagnetic Field Strength Reduction Possibilities**

Author **Saulius Dreinius**

Academic supervisor **Assoc Prof Dr Ričardas Visvaldas Pocius**

Thesis language:
Lithuanian

Annotation

This final study is based on the research of the reduction possibilities of the wireless communication network base stations generating electromagnetic field strength in close environment. The parameters of the base stations antennas and the valid hygiene standards are also described. The methodology which enables to calculate the electromagnetic field strength generated by antenna is studied here. Moreover, experimental EML intensity research and its results are analyzed in this final study. The special developed program, simulating the radiation intensity in close environment and indicating fixed prohibited zones on the map, is also introduced. This program is capable to select parameters automatically in pursuance of reducing the radiation intensity in chosen points. After having performed simulations, the parameters are discussed. There is the possibility to reduce EML energy power density by changing these parameters.

Work size - 53 p. text without appendixes, 31 illustrations, 21 bibliographic sources. Appendixes included.

Keywords: electromagnetic field strength, power density, EMF reduction, program simulating the radiation intensity

Vilniaus Gedimino technikos universiteto egzaminų
sesijų ir baigiamųjų darbų rengimo bei gynimo
organizavimo tvarkos aprašo 2011–2012 m. m.
2 priedas

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Saulius Dreinius, 20074812

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Elektronikos

(Fakultetas)

Telekomunikacijų inžinerija, TETfm-11

(Studijų programa, akademinė grupė)

**BAIGIAMOJO DARBO (PROJEKTO)
SĄŽININGUMO DEKLARACIJA**

20 m. _____ d.
(Data)

Patvirtinu, kad mano baigiamasis darbas (projektas) tema Bevielių ryšio tinklų bazinių
stočių sukuriamų elektromagnetinių laukų intensyvumo sumažinimo galimybių tyrimas

patvirtintas 20 ____ m. _____ d. dekanu potvarkiu Nr. _____, yra
savarankiškai parašytas. Šiame darbe (projekte) pateikta medžiaga nėra plagijuota. Tiesiogiai ar
netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Prenkant ir įvertinant medžiagą bei rengiant baigiamąjį darbą (projektą), mane konsultavo
mokslininkai ir specialistai: _____

Mano darbo (projekto) vadovas doc. dr. Ričardas Visvaldas Pocius

Kitų asmenų indėlio į parengtą baigiamąjį darbą (projektą) nėra. Jokių įstatymų nenumatytų
piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).

(Parašas)

(Vardas ir pavardė)

TURINYS

TURINYS.....	7
1. ĮVADAS. UŽDUOTIES ANALIZĖ	9
1.1. Darbo tikslas.....	9
1.2. Darbo uždaviniai	10
1.2. Tyrimų metodai	10
1.3. Darbo naujumas ir praktinė nauda	10
1.4. Darbo struktūra.....	10
2. EML VERTINIMO BŪDAI	11
2.1. Mobiliojo ryšio tinkas	11
2.2. Anteninių įrenginių savybės.....	13
2.3. EML energijos srauto tankis	15
2.4. EML energijos srauto tankį reglamentuojantys dokumentai.....	16
2.5. EML energijos srauto tankio skaičiavimo metodika.....	18
2.6. Okumura-Hata empirinis modelis	20
2.7. Išvados.....	22
3. EML VERTINIMO MODELIŲ REALIZACIJA	23
3.1. Tyrime naudota antena	23
3.2. Tyrimas, eiga ir gauti rezultatai.....	24
3.3. Išvados.....	27
4. EML ĮVERTINIMO PROGRAMOS KŪRIMAS	28
4.1. Programos veikimo aprašymas.....	28
4.2. Programos skaičiavimų patikrinimas	30
4.3. Išvados.....	30
5. EML MINIMIZAVIMO GALIMYBIŲ TYRIMAS.....	32
5.1. Energijos srauto tankio modeliavimo parametrai.....	32
5.2. Antenos išspinduliuojamos galios įtaka energijos srauto tankiui	33
5.3. Antenos bei matavimo aukščio įtaka energijos srauto tankiui	35
5.4. Gretimų antenų įtaka energijos srauto tankiui.....	38
5.5. Antenų palenkimo ir pasukimo įtaka energijos srauto tankiui.....	40
5.6. Energijos srauto tankio minimizavimo galimybės	42
5.6. Išvados.....	47

6. IŠVADOS.....	50
7. LITERATŪROS SĄRAŠAS.....	52
A PRIEDAS. PROGRAMOS NAUDOJIMO APRAŠYMAS	54
A.1. Programos įrankių juosta.....	54
A.2. Žemėlapių navigacija	55
A.3. Žemėlapių sluoksniai.....	55
A.4. Programos nustatymai	56
A.5. Radiotechniniai objektai.....	57
A.6. Duomenų įkrovimas ir išsaugojimas	58
A.7. Rezultatų modeliavimas	59
A.8. Gauti rezultatai	59
A.9. Papildomi įrankiai	60
A.10. Išvados.....	60
B PRIEDAS. ELEKTROMAGNETINIO LAUKO INTENSYVUMO MATAVIMO AKTAS...	61
C PRIEDAS. PROGRAMOS IŠEITIES KODAS	62
D PRIEDAS. PAŽYMA APIE DALYVAVIMĄ XVI-OJE LIETUVOS JAUNŲJŲ MOKSLININKŲ KONFERENCIJOJE „MOKSLAS – LIETUVOS ATEITIS“	88

1. ĮVADAS. UŽDUOTIES ANALIZĖ

Lietuvoje įdiegiama vis daugiau naujų mobiliojo ryšio bazinių stočių. Plėtojami nauji tinklai, plečiami jau esami. Didėjant belaidės elektroninės įrangos naudojimui, griežtėja ir elektromagnetinės spinduliuotės normos. Paslaugos operatorius, nusprendęs norimoje vietoje įrengti radiotechninį objektą stengiasi, kad būtų užtikrinama paslaugos kokybė dideliame plote. Vartotojai pageidauja geros ryšio kokybės ir didelės greitaveikos interneto, tuo pačiu prieštaraudami didesnio intensyvumo nejonizuojančiai spinduliuotei.

Daugiausia problemų įrengiant minėtas stotis kyla miestuose ten, kur šalimais jau veikia kiti radiotechniniai objektai. Bendra radijo bangas spinduliuojančių objektų visuma sukuria intensyvią elektromagnetinį lauką (EML), kurį Lietuvoje griežtai reglamentuoja higienos normoje HN 80:2011 nustatyti leidžiami lygiai. Reikalaujama, kad kiekvieno naujai įrengiamo ar atnaujinamo objekto sukuriama spinduliavimo intensyvumas būtų įvertinamas ir neviršytų nustatyto $10 \mu\text{W}/\text{cm}^2$ elektromagnetinio lauko energijos srauto tankio [1].

Elektromagnetines bangas spinduliuojančius objektus galima įvertinti tiek eksperimentiškai, tiek ir naudojant teorinius skaičiavimo metodus. Teorinis spinduliavimo intensyvumo vertinimas Lietuvoje nėra griežtai apibrėžtas, todėl derinant radiotechninio objekto radiotechninės dalies projektą reikalaujama nurodyti skaičiavimo metodiką [2]. Priimta, kad vertinant antenos sukuriamą spinduliavimo intensyvumą naudojama jau nusistovėjusi metodika EML stipriui apskaičiuoti [3].

Daugėjant elektromagnetinį lauką sukuriančių objektų skaičiui didėja ir bendras spinduliavimo intensyvumo lygis. Svarbus uždavinys įrengiant šiuos objektus ne tik neviršyti leidžiamų verčių bet ir stengtis, kad objektas sukeltų kuo mažesnę spinduliavimo intensyvumą artimoje aplinkoje. Tai įgyvendinama ne tik keičiant siųstuvo darbo režimą, bet ir tinkamai parenkant antenos modelį bei montavimo parametrus.

1.1. Darbo tikslas

Šio darbo tikslas ištirti bevielų ryšio tinklų bazinių stočių sukuriamos elektromagnetinės spinduliuotės intensyvumo sumažinimo galimybes artimoje siųstuvų aplinkoje plėtojant egzistuojančius bevielio ryšio tinklus ir diegiant naujas technologijas.

1.2. Darbo uždaviniai

Darbo tikslui pasiekti formuluojami šie uždaviniai:

- 1) išnagrinėti elektromagnetinių laukų intensyvumui vertinti skirtus modelius;
- 2) atlikti realios elektromagnetinės situacijos eksperimentinį tyrimą;
- 3) sukurti programą, kuri sumodeliuotų spinduliavimo intensyvumą artimoje aplinkoje, nustatytas draudžiamąsias zonas atvaizduotų žemėlapyje;
- 4) įvertinti kokiomis priemonėmis galima sumažinti EML energijos srauto tankį artimoje aplinkoje.

1.2. Tyrimų metodai

Eksperimentinis matavimas atliktas EML matavimo prietaisu, modeliavimų rezultatai apskaičiuoti pasinaudojant sukurta programine įranga. Radiotechninių objektų sukuriama elektromagnetinės spinduliuotės intensyvumo vertinimas atliekamas pagal Lietuvoje naudojamą modelį.

1.3. Darbo naujumas ir praktinė nauda

Pastaruoju metu atlikta daug darbų, kuriuose matuojamas ir vertinamas bazinių stočių sukuriama EML energijos srauto tankis. Tačiau iki šiol dar nėra detaliai aptarta, kokią įtaką spinduliavimo intensyvumui turi įvairūs antenų parametrai. Nėra ištirta, kaip keičiasi energijos srauto tankis artimoje aplinkoje keičiant daugiau nei vieną aptartą parametą. Svarbu, kad sumažinus spinduliavimą artimoje aplinkoje būtų ir toliau užtikrinama paslaugos kokybė norimu spinduliu.

Darbe sukurta programa, padėsianti automatizuoti spinduliavimo intensyvumo skaičiavimus. Įdiegta galimybė, automatiškai keisti parametrus, siekiant sumažinti spinduliavimo intensyvumą pasirinktuose taškuose. Atlikus modeliavimus aptariama keičiamų parametų įtaka EML energijos srauto tankiui bei padengimui norimu spinduliu.

1.4. Darbo struktūra

Magistro baigiamąjį darbą sudaro septyni skyriai. Darbo apimtis yra 53 puslapiai be priedų, tekste yra 31 numeruotas paveikslas. Naudojamas 21 literatūros šaltinis.

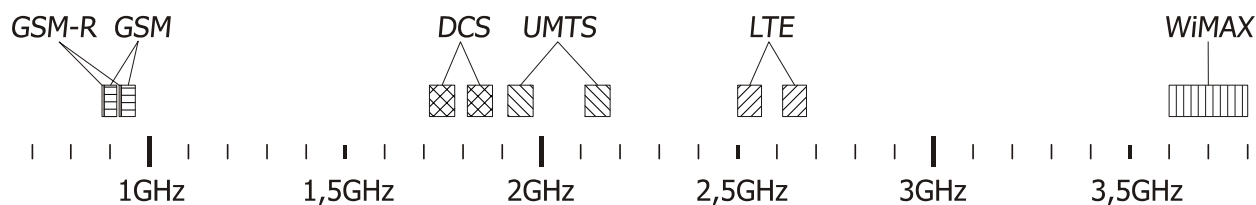
2. EML VERTINIMO BŪDAI

Skyriuje aptariami Lietuvoje veikiančių mobiliojo ryšio technologijų dažniai. Trumpai peržvelgiama bazinių stočių veikimo bei jų dažnių planavimo metodika. Išnagrinėjamos anteninių įrenginių savybės.

Apžvelgiami energijos srauto tankį Lietuvoje reglamentuojantys dokumentai bei Europos Sąjungos rekomendacija. Nagrinėjama Lietuvoje naudojama EML energijos srauto tankio skaičiavimo metodika.

2.1. Mobiliojo ryšio tinkas

Lietuvoje pagrindinėms mobiliojo ryšio technologijoms išskirtų diapazonų išsidėstymas pavaizduotas 2.1 pav. GSM-R skirta 876–880 MHz terminalų siuntimui ir 921–925 MHz bazinių stočių (BS) siuntimui. GSM — 880–915 MHz, bei 925–960 MHz. 1710–1785 MHz ir 1805–1880 MHz — DCS technologijai. UMTS išskirta 1920–1980 MHz ir 2110–2170 MHz. Naujausiajai LTE technologijai suteikė 2500–2560 MHz ir 2620–2680 MHz diapazonus. Mobiliam internetui *WiMax* skirta 3600–3800 MHz diapazonas [4].



2.1 pav. Mobiliojo ryšio technologijos dažnių ašyje

Mobiliojo ryšio bazinės stotys yra sujungtos į bendrą tinklą. Kiekviena BS savo ląstelėje gali užmegzti ryšį su keliais terminalais. Komunikavimui tarp BS ir terminalo reikalingi radijo dažniai. Kadangi radijo dažnių ištekliai yra riboti, dėl ląstelių juos galima panaudoti pakartotinai [5].

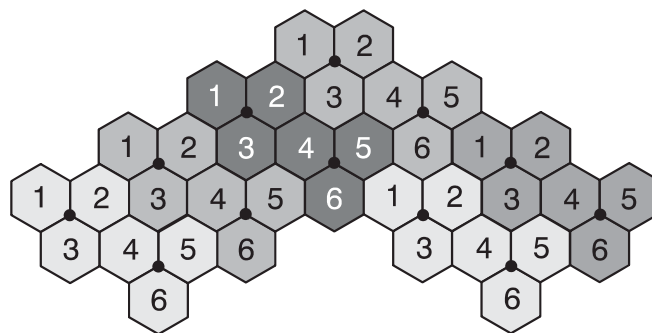
Ląstelių padengiamas plotas skiriasi — vienos padengia kelių šimtų metrų spinduliu miestuose, kitos — keliolikos kilometrų spinduliu užmiestyje. Taip pat naudojamos ir mažosios ląstelės, kuriomis mažame plote galima padidinti sistemos talpą ar pagerinti ryšio kokybę.

Kiekviena BS operuoja keliais kanalais vienu metu, kiekvienas kanalas turi dažnius ryšiui BS–terminalas ir terminalas–BS. Vienas kanalas gali aptarnauti vieną arba kelis terminalus — priklausomai nuo sistemos. Šiuo metu Lietuvoje neveikianti NMT sistema vienam terminalui

aptarnauti išskirdavo vieną kanalą, kurio juostos plotis 25 kHz. Naujesnė, antros kartos sistema viename, 200 kHz juostos pločio, kanale sukuria 8 laiko tarpsnius. Vieną iš jų, paprastai, naudoja kontroliniam kanalui siųsti, o likusius — terminalams aptarnauti. Trečiosios kartos sistemoje įgyvendintas kodinis sutankinimas įgalina 5 MHz juostos pločio kanale aptarnauti iki 100 terminalų vienu metu [6].

Kiekviena bazinė stotis visada siunčia vieną pastovios galios kontrolinį pokanalį. Likę pokanaliai naudojami esant poreikiui — reikalingas kiekis pokanalių aktyvuojamas tada, kai inicijuojamas sujungimas su terminalu. Duomenims ir balsui skirtų pokanalių galia lanksčiai reguliuojama [7], yra sukurti algoritmai, kurie aptinka tylą pokalbyje ir tam laiko momentui siuntimą nutraukia. Reikia pastebėti, jog net ir esant pilnam BS apkrovimui nebūtinai bus išspinduliuojama maksimali galia.

Projektuojant antrosios kartos bazinių stočių išsidėstymą stengiamasi kuo optimaliau išnaudoti operatoriaus turimą radijo dažnių spektrą. Kadangi naudojamų kanalų skaičius yra ribotas, mobiliojo ryšio tinkle tuo pačiu dažniu veikia daug bazinių stočių. Patikimą tokios sistemos veikimą galima užtikrinti tik tinkamai valdant bazinių stočių spinduliavimo galią bei antenų palenkimą. Kitu atveju tikėtini tarpkanaliniai trukdžiai artimiausiose, tais pačiais dažniais dirbančiose BS (2.2 pav.).



2.2 pav. Tinklo klasteriai, kurių pakartotinio panaudojimo skaičius — 6

Trečiosios kartos judriojo korinio ryšio sistemose nebereikia planuoti tinklo ląstelių darbinio dažnių, tačiau vis tiek išlieka būtinybė griežtam signalų galios valdymui. BS signalo spinduliavimo galia turėtų būti pakankamai didelė, norint užtikrinti norimą padengiamumą bei paslaugos kokybę, ir pakankamai maža, norint išvengti interferencijos. Tankiai apgyvendintose vietose, tinklo talpa didinama plečiant BS skaičių. Daugėjant BS skaičiui, mažėja reikalingi aptarnauti plotai, dėl to galima sumažinti spinduliuojamą galią.

Plotas, kurį turės aptarnauti bazinė stotis, padalijamas į tris sektorius. Įprastai kiekvienam sektoriui skirtos atskiros BS ir antena. Vietose, kur poreikis tinklo talpai didesnis, vieno sektoriaus BS turi kelis kanalus, kurių signalai sumuojami ir išspinduliuojami per tą pačią sektorinę anteną. Naudojamos ir sektorinės antenos, kurios skirtos kelioms technologijoms — tokiu atveju prie vienos tokios antenos gali būti prijungiama skirtingų technologijų bazinės stotys, kurios turi po kelis kanalus. Plotuose, kuriuose ląstelės skirstyti į sektorius nereikšminga — naudojamos visakryptės antenos.

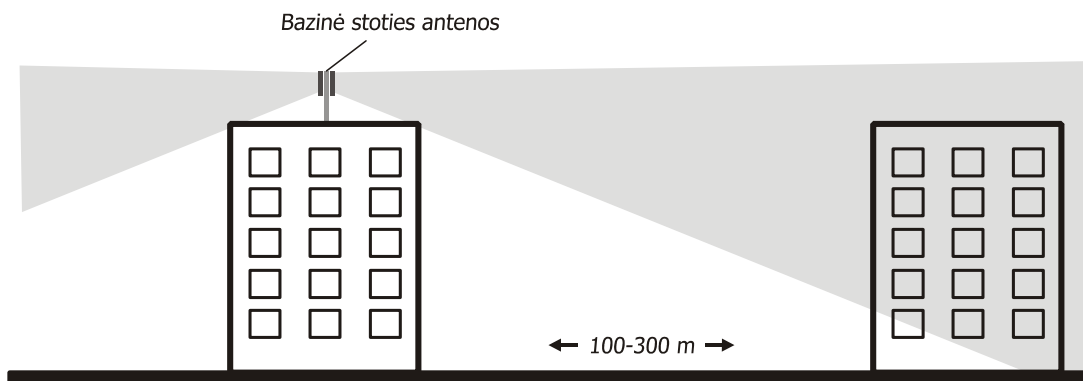
2.2. Anteninių įrenginių savybės

Bazinių stočių sukuriama radijo signalai išspinduliuojami per antenas suformuojant kryptingus spindulius. Spindulio profilis parenkamas tinklo planuotojų tam, kad būtų užtikrintas optimalus reikiamo ploto padengimas.

Antenos sukuriamas elektromagnetinis laukas priklauso nuo signalo bangos ilgio, siųstuvo galios ir antenos parametrų: stiprinimo koeficiento, įėjimo varžos, krypties efektyvumo koeficiento, kryptingumo diagramos, praleidžiamų dažnių juostos, efektyviojo ilgio ir stovinčios bangos koeficiento.

Didelius plotus aptarnaujančios antenos turi siaurą vertikalaus spinduliavimo diagramą, paprastai 5° – 10° . Antena sumontuojama taip, kad spindulys būtų nukreiptas arčiau žemės — tai vadinama palenkimu. Tokios antenos pagrindinio lapelio spindulys žemę pasiekia už 50–200 m priklausomai nuo sumontavimo aukščio. Tai paaiškina reiškinį, kai signalo galia visai šalia, po antena, būna daug mažesnė nei už kelių šimtų metrų. Pasinaudojant šia ypatybe viršutiniuose namo aukštuose užtikrinamas pakankamai žemas elektromagnetinis laukas, kai ant stogo yra įrengiama antena (2.3 pav.).

Paprastai kryptingumo diagrama charakterizuojama kampu, kuriame antenoje kuriamas EML nesumažėja daugiau kaip 3 dB nuo didžiosios reikšmės [8].



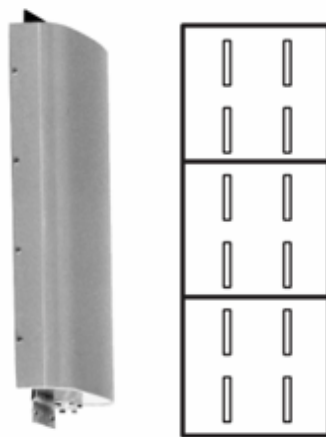
2.3 pav. Sektorinės antenos pagrindinio lapelio spinduliavimas

Vertikaliam antenos palenkimui gali būti naudojamas mechaninis palenkimas, kai pati antena fiziškai palenkiamą žemyn. Taip pat naudojamas ir elektrinis palenkimas, kai nuosvyrio kampas reguliuojamas keičiant atskirų masyvo spinduolių išspinduliuojamų signalų fazes.

Siaura vertikali antenos spinduliavimo diagrama susidaro dėl antenoje esančių techninių sprendimų — vertikaliame masyve išdėstytų spinduliuojančių elementų (2.4 pav.). Kuo daugiau vertikaliai išdėstytų elementų masyve — tuo siauresnė vertikali spinduliavimo diagrama.

Horizontalios diagramos spinduliavimą (120° – 60°) nulemia sumontuotas reflektorius. Reflektoriaus sukuriama priekinio ir atgalinio spinduliavimo intensyvumo santykis yra apie 30 dB. Šis parametras naudojamas kontroliuoti kanalų pakartotinio panaudojimo galimybes.

Antenos stiprinimo koeficientas nusako, kiek kartų konkrečios antenos kuriamas elektromagnetinis laukas didesnis už elektromagnetinį lauką pusbangio vibratoriuje, kai antenos ir vibratoriaus signalo siuntimo sąlygos vienodos ir jie optimaliai suderinti su siųstuvo išėjimu. Stiprinimo koeficientas nulemia ir antenos krypties savybes: kuo didesnis stiprinimo koeficientas, geresnės antenos krypties savybės, tuo siauresnė kryptingumo diagramoje svarbiausios krypties juosta. Paprastai bazinių stočių mobiliojo ryšio antenų stiprinimas yra 10-20 dB.



2.4 pav. Sektorinė antena ir vertikalių spinduolių išdėstymas joje [9].

Stovinčios bangos koeficientas — tai dydis, nusakantis antenos įėjimo varžos ir fiderio banginės varžos santykį arba suderinimo lygį. Jeigu šios varžos nesuderintos, mažėja sukuriama elektromagnetinis laukas, o perdavimo linijoje atsiranda signalo atspindžių.

Lastelėje, kurioje reikia užtikrinti kelių mobiliojo ryšio technologijų veikimą montuojamos kelių diapazonų antenos. Pastarosios įgyvendintos pasinaudojant X poliarizacijos antenos prototipu, kai keli skirtingų ilgių masyvo spinduoliai sumontuojami vienos antenos plokštumoje (2.4 pav.). Tokioje užtikrinamas 30 dB izoliavimas tarp skirtingų diapazonų.

Koks bus antenos sukuriamas elektromagnetinio lauko stiprumas priklauso nuo BS siųstuvo galios bei antenos stiprinimo. Šių parametų suma nusakoma kaip efektyvioji izotropinė spinduliuojamoji galia (EIRP), matuojama decibelais pagal milivatą (dBm). Apskaičiuojama sumuojant antenos stiprinimą bei siųstuvo galią decibelais.

Gretimai mobiliojo ryšio antenų įrengiamos ir duomenų perdavimui skirtos antenos. Sistema skirta sujungti BS į vieną tinklą vadinama radijo rėline linija. Jai įgyvendinti naudojamos parabolinės antenos užtikrinančios ryšį taškas–taškas. Lietuvoje mobiliųjų tinklų operatoriai naudoja 6–38 GHz diapazone veikiančius duomenų perdavimo įrenginius.

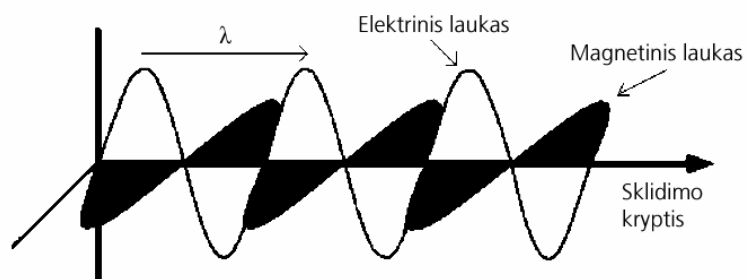
Parabolinės antenos sukuria siaurą kryptingumo diagramą, kuri siekia tik 1–2°. Siųstuvų galia paprastai būna ne didesnė kaip 25 dBm, antenų stiprinimas 25–45 dB.

2.3. EML energijos srauto tankis

Spinduliavimas yra apibrėžiamas kaip energijos sklidimas erdvėje bangų arba dalelių forma. Elektromagnetinę spinduliuotę galima nusakyti kaip visumą elektrinių ir magnetinių bangų, judančių erdvėje. Šios bangos yra sukuriamos elektrinių srovių judėjimo spinduolyje. Pvz. kintamosios srovės judėjimas antenoje generuoja elektromagnetines bangas, kurios sklinda tolyn nuo siųstuvo antenos.

Terminas „elektromagnetinis laukas“ naudojamas nusakyti elektromagnetinės energijos buvimą tam tikroje vietoje. Šis laukas gali būti aprašytas elektrinio ir/ar magnetinio lauko stipriu.

Dėl to, kad elektromagnetinis laukas turi elektrinę ir magnetinę dedamąsias, dažnai stiprumą paranku išreikšti atskirais vienetais kiekvienai dedamajai. Voltais viename metre (V/m) išreiškiamas elektrinio lauko stiprumas, o amperais viename metre (A/m) nurodomas magnetinio lauko stiprumas.



2.5 pav. Elektrinio ir magnetinio lauko dedamosios

Kitas dažnai naudojamas matavimo vienetas nusakyti elektromagnetinio lauko stiprį yra energijos srauto tankis. Šis dydis naudojamas, kai matavimo taškas yra gana toli nuo siųstuvo

antenos. Pastarasis atstumas yra bent keli bangos ilgiai nuo spinduliuotės šaltinio ir vadinamas tolimuoju lauku. Tolimajame lauke elektrinis ir magnetinis laukas yra aiškiai tarpusavyje susiję (2.5 pav.), todėl užtenka išmatuoti vienos dedamosios stiprį, tam kad nusakyti likusios vertę.

Pointingo teorema nusako ryšį tarp energijos srauto tankio (P_d), elektrinio lauko (E) ir magnetinio lauko (H) vektorių [10]:

$$P_d = E \cdot H \quad [W/m^2] \quad (2.1)$$

Artimojo lauko zonoje fizinis ryšys tarp elektrinės ir magnetinės dedamosios laukų įprastai yra sudėtingas. Šiuo atveju, būtina nustatyti tiek elektrinio tiek ir magnetinio laukų stiprius tam, kad charakterizuoti aplinką. Energijos srauto tankis apibūdinamas kaip galia ploto vienetu. Pvz. gali būti išreiškiama milivatais kvadratiniam centimetre (mW/cm^2) ar mikrovatais kvadratiniam centimetre ($\mu W/cm^2$).

2.4. EML energijos srauto tankį reglamentuojantys dokumentai

Europos sąjungos tarybos rekomendacijoje priimtoje 1999 m. liepos 12 d. (1999/519/EB) yra išdėstyti pagrindiniai gyventojų apsaugos nuo EML poveikio nuostatai. Rekomendacijoje nurodyti pagrindiniai apribojimai ir ribinės vertės taikomi bazinėms mobiliojo ryšio stotims. Šie lygiai yra grindžiami 1998 m. Tarptautinės apsaugos nuo nejonizuojančiosios spinduliuotės komisijos (International Commission for Non-ionising Radiation Protection, ICNIRP) gairėmis [11], parengtomis atsižvelgiant į trumpalaikį EML poveikį. Antroje šios ES rekomendacijos ataskaitoje 2007 m. pranešama, kad dauguma valstybių narių priėmė rekomendaciją [12].

Rekomendacijoje nurodoma, jog dažniuose, kuriuose veikia mobiliojo ryšio bazinės stotys savitosios energijos absorbuojimo rodiklis (SAR) visam žmogaus kūnui negali viršyti 0,08 W/kg. Esant didesnėms vertėms fiksuojamas fiziologiškai svarbus žmogaus kūno temperatūros padidėjimas. Pagal šį rodiklį nustatyta išvestinė ribinė EML energijos srauto tankio vertė, kuri apskaičiuojama pagal formulę $f/200$ ir tiesiogiai priklauso nuo dažnio [13]. Naudojantis nurodyta priklausomybe nuo dažnio gautos ribinės vertės Lietuvoje veikiančioms technologijoms — GSM 450 $\mu W/cm^2$, DCS 900 $\mu W/cm^2$, UMTS 1000 $\mu W/cm^2$, LTE 1300 $\mu W/cm^2$, WiMAX 1850 $\mu W/cm^2$.

Lietuvos galioja higienos norma HN 80:2011 „Elektromagnetinis laukas darbo vietose ir gyvenamojoje aplinkoje. Parametrų normuojamos vertės ir matavimo reikalavimai 10 kHz – 300 GHz radijo dažnių juostoje“. Ji nustato radiotechniniams objektams taikomus visuomenės sveikatos saugos reikalavimus, elektromagnetinio lauko intensyvumo parametrų leidžiamas vertes bei matavimo reikalavimus gyvenamojoje aplinkoje ir darbo vietose, kuriose įrengti elektromagnetinės spinduliuotės šaltiniai, bei nurodo apsaugos priemones, skirtas mažinti elektromagnetinės

spinduliuotės poveikį darbuotojų sveikatai. Šios higienos normos laikytis privaloma visiems Lietuvos Respublikos ar kitos Europos Sąjungos valstybės narės piliečiams, kitiems fiziniams asmenims, kurie projektuoja didesnės negu 25 W efektyviosios spinduliuotės galios radiotechninius objektus ūkinei komercinei veiklai vykdyti, atlieka šių objektų elektromagnetinės spinduliuotės matavimus, skaičiavimus, stato (įrengia) šiuos objektus ar juos eksploatuoja arba eksploatuos (operatorius). Kadangi radijo dažnių juostoje nuo 300 MHz iki 300 GHz normuojamas energijos srauto tankis S , tai didžiausias galimas intensyvumas yra $10 \mu\text{W}/\text{cm}^2$ [1].

Radiotechninio objekto radiotechninės dalies projekto ir elektromagnetinės spinduliuotės stebėsenos plano derinimo tvarkos aprašas nustato radiotechninio objekto radiotechninės dalies projektui derinti reikalingų dokumentų ir duomenų pateikimo, jų vertinimo ir sprendimų priėmimo tvarką bei elektromagnetinės spinduliuotės stebėsenos plano sudarymo, dokumentų ir duomenų, reikalingų elektromagnetinės spinduliuotės stebėsenos plano derinimui, pateikimo, jų vertinimo, koregavimo, sprendimų priėmimo bei elektromagnetinės spinduliuotės stebėsenos plano vykdymo tvarką. Šis aprašas kaip ir higienos norma HN 80:2011 privaloma visiems, kurie projektuoja didesnės negu 25 W efektyviosios spinduliuotės galios radiotechninius objektus. Radiotechninės dalies projektą operatoriai privalo derinti prieš įrengdami radiotechninį objektą arba kai keičiama radiotechninio objekto antenų įrengimo vieta, jų aukštis virš žemės paviršiaus, didžiausios spinduliuotės kryptis arba kai didinama radiotechninio objekto efektyviosios spinduliuotės galia bet kuria kryptimi. Elektromagnetinės spinduliuotės parametrų pasiskirstymo skaičiavimai siekiant nustatyti, kokiais atstumais nuo antenų ir kokiame aukštyje virš žemės paviršiaus pasiekiamos ribinės vertės, atliekami nuo 0° azimuto kas 10° , $1,5 \pm 0,2$ m aukštyje virš žemės paviršiaus, artimiausio gyvenamojo ar visuomeninės paskirties pastato viršutinio eksploatuojamo aukšto langų centro lygyje ir $1,5 \pm 0,2$ m aukštyje virš šio pastato stogo, viršutinio eksploatuojamo aukšto langų centro lygyje ir $1,5 \pm 0,2$ m aukštyje virš pastato stogo, ant kurio bus įrengtos antenos, kai antenos projektuojamos ant pastato stogo, pastato, ant kurio sienos bus įrengtos antenos, arčiausiai esančių langų centro lygyje ir antenų įrengimo lygyje, kai antenos projektuojamos ant pastato sienos. Vertinimo metu nustačius, kad radiotechninio objekto radiotechninės dalies projektas neatitinka nustatytų reikalavimų priimamas sprendimas nederinti radiotechninio objekto radiotechninės dalies projekto [2]. Skaičiavimai parodė, jog atlikti EML energijos srauto tankio tyrimus didesniu kaip 300 m atstumu nėra reikalinga, nes esant tokiam atstumui bazinės stoties sukurtas intensyvumas yra labai mažas [14].

2.5. EML energijos srauto tankio skaičiavimo metodika

Teorinis spinduliavimo intensyvumo vertinimas Lietuvoje nėra griežtai apibrėžtas, todėl derinant radiotechninio objekto radiotechninės dalies projektą reikalaujama nurodyti skaičiavimo metodiką [2]. Priimta, kad vertinant antenos sukuriamą spinduliavimo intensyvumą naudojama jau nusistovėjęs metodika EML stipriui apskaičiuoti [3], kuri oficialiai patvirtinta Rusijoje [15].

Poskyryje aptariamas modelis tinka apskaičiuoti EML stiprį artimojoje siųstuvų aplinkoje, kai siųstuvai veikia 27 – 2400 MHz diapazone. Modelyje įvertinamas slopinimas laisvoje erdvėje, o skaičiavimas grindžiamas pačių nepalankiausių aplinkybių įvertinimu. Todėl yra taikomi atsargos koeficientai dėl antenų kryptingumo diagramų netolygumo ir galimų žemės paviršiaus savybių pokyčių esant skirtingoms meteorologinėms sąlygoms [16]. Skaičiavimai atliekami naudojantis tokia išraiška:

$$E = \frac{\sqrt{30 \cdot P \cdot G \cdot \eta_f}}{D} \cdot F(\Delta) \cdot F(\varphi) \cdot K_Z \cdot K_H \quad (2.1)$$

čia: E — elektrinio lauko stipris (V/m); P — siųstuvo galia (W); G — antenos stiprinimo koeficientas, kartais pagal dBd; η_f — siuntimo sistemos kokybės koeficientas; D — atstumas nuo antenos geometrinio centro iki skaičiuojamo taško (m); $F(\Delta)$ — vertikalioje plokštumoje normalizuotas kryptingumo diagramos dydis; $F(\varphi)$ — horizontalioje plokštumoje normalizuotas kryptingumo diagramos dydis; K_Z — žemės paviršiaus įtakos bangų sklidimui koeficientas; K_H — nukrypimo nuo pagrindinės antenos spinduliavimo krypties horizontalioje plokštumoje koeficientas.

Apskaičiuotas elektrinio lauko stipris gali būti perskaičiuotas į EML energijos srauto tankį naudojantis santykiu tarp elektrinio lauko stiprio ir varžos laisvoje erdvėje (120π):

$$S = \frac{E^2}{377} \quad (5.2)$$

čia: S — energijos srauto tankis (W/m²), E — elektrinio lauko stipris (V/m).

Antenos išspinduliuota energija priklauso nuo antenos stiprinimo, siųstuvo galios ir siuntimo sistemos kokybės. Perdavimo linijos nuostolių ir antenos kokybės koeficiento sandauga yra lygi siuntimo sistemos kokybės koeficientui.

Perdavimo linijos koeficientas apskaičiuojamas žinant linijos kokybę ir jos ilgį:

$$\eta_k = 10^{-\frac{\alpha \cdot l}{10}} \quad (5.3)$$

čia: η_k — perdavimo linijos nuostolių koeficientas; α — kabelio slopinimas (dB/m); l — kabelio ilgis (m).

Antenos kokybės koeficientą apskaičiuojamas žinant antenos stovinčios bangos koeficientą:

$$\eta_a = 1 - \left(\frac{K_c - 1}{K_c + 1} \right)^2 \quad (5.4)$$

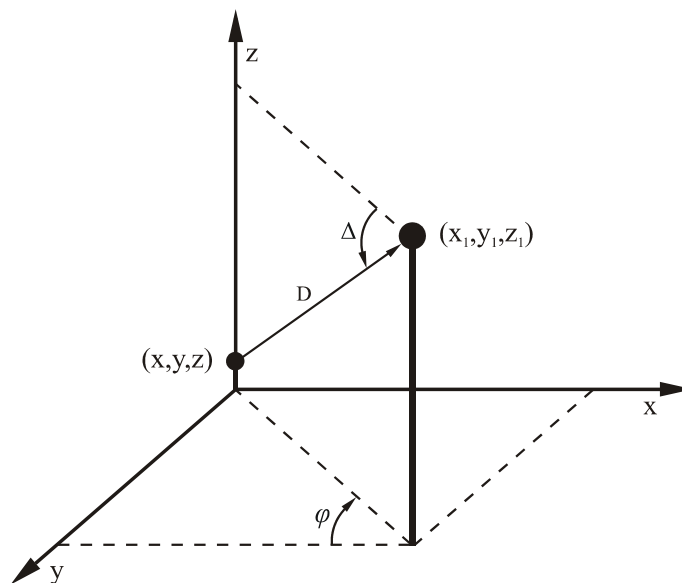
čia: η_a — antenos kokybės koeficientas; K_c — antenos stovinčios bangos koeficientas.

Spinduliavimo intensyvumo slopinimas tiriamajame taške priklauso nuo atstumo iki antenos geometrinio centro bei antenos kryptingumo charakteristikos apskaičiuotais kampais vertikalioje ir horizontalioje plokštumose (2.6 pav.).

Atstumas nuo antenos geometrinio centro iki tiriamojo taško apskaičiuojamas pagal:

$$D = \sqrt{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2} \quad (5.5)$$

čia: D — atstumas (m); x, y, z — tiriamojo taško padėtis koordinačių sistemoje; x_1, y_1, z_1 — antenos padėtis koordinačių sistemoje.



2.6 pav. Kampai ir atstumai nuo tiriamojo taško iki antenos

Kampas, kurį su antenos plokštuma sudaro tiesė nubrėžta iki tiriamojo taško, gaunamas pagal formulę:

$$\Delta = \arctan \left(\frac{z_1 - z}{\sqrt{(x_1 - x)^2 + (y_1 - y)^2}} \right) \quad (5.6)$$

čia: Δ — kampas vertikaliajoje plokštumoje.

Kampas, tarp antenos intensyviausios spinduliavimo krypties ir tiriamojo taško apskaičiuojamas pagal formulę:

$$\varphi = \arctan \left(\frac{y_1 - y}{x_1 - x} \right) \quad (5.7)$$

čia: φ — kampas horizontalioje plokštumoje.

Normalizuoti kryptingumo diagramos dydžiai gaunami perskaičiavus kryptingumo diagramos slopinimą:

$$F = 10^{-\frac{k}{20}} \quad (5.8)$$

čia: F — normalizuotas kryptingumo diagramos dydis; k — kryptingumo diagramos slopinimas (dB).

Modeliavimų metu gautus rezultatus lyginant su išmatuotais buvo parinktas antenų kryptingumo diagramų netolygumo ir galimų žemės paviršiaus savybių pokyčių koeficientas, lygus 1,63.

2.6. Okumura-Hata empirinis modelis

Okumura-Hata empirinis modelis yra plačiai naudojamas sklaidimo slopinimui apskaičiuoti didelėse ląstelėse. Okumura atliko bandymus kelių dydžių Japonijos miestuose išmatuodamas signalo slopinimą esant skirtingam antenų aukščiui ir matavimo atstumui [17]. Hata perskaičiavo Okumura gautus duomenis į matematinės išraiškas [18]. Modelyje įvesti bazinės stoties bei terminalo antenų aukščių, taip pat matavimo atstumo koeficientai. Šis modelis plačiai panaudojamas dėl skirtingo dydžio miestams išvestų koeficientų.

Skaičiavimai naudojami 150 – 1500 MHz diapazone, kai bazinės stoties antenos aukštis kinta nuo 30 m iki 200 m, o terminalo antenos aukštis nuo 1 m iki 10 m. Maksimalus atstumas kuriame galima įvertinti signalo slopinimą yra 20 km, tačiau ne mažiau kaip 1 km. Dėl dažnio apribojimo

yra įgyvendintas modelio papildymas skaičiavimams iki 2000 MHz, kuris pavadintas COST 231 [19].

Pagrindinė modelio išraiška, kurios pagalba galima apskaičiuoti kelio nuostolius:

$$PL = A + B \log(d) + C \quad (5.10)$$

čia: PL — kelio nuostoliai (dB); A , B , C — slopinimo koeficientai; d — atstumas tarp bazinės stoties ir imtuvo antenos (km).

Slopinimo koeficientai apskaičiuojami pagal išraiškas:

$$A = 69,55 + 26,16 \log(f_c) - 13,82 \log(h_b) - a(h_m) \quad (5.11)$$

$$B = 44,9 - 6,55 \log(h_b) \quad (5.12)$$

čia: f_c — siųstuvo dažnis (MHz); h_b — siųstuvo antenos aukštis (m); h_m — imtuvo antenos aukštis (m); $a(h_m)$ — korekcijos koeficientas skirtingo tipo keliui.

Koeficientas dideliems miestams (*metropolitan areas*), $C = 0$:

$$a(h_m) = \begin{cases} 8,29(\log(1,54h_m))^2 - 1,1 & \text{kai } f \leq 200\text{MHz} \\ 3,2(\log(11,75h_m))^2 - 4,97 & \text{kai } f \geq 400\text{MHz} \end{cases} \quad (5.13)$$

Koeficientas vidutinio dydžio miestams (*small and medium-size cities*), $C = 0$:

$$a(h_m) = (1,1 \log(f_c) - 0,7)h_m - (1,56 \log(f_c) - 0,8) \quad (5.14)$$

Atliekant skaičiavimus priemiesčio (*suburban*) rajonuose naudojamas vidutinio dydžio miestams skirtas koeficientas su papildomu slopinimo koeficientu:

$$C = -2 \left(\log \left(\frac{f_c}{28} \right) \right)^2 - 5,4 \quad (5.15)$$

Mažo užstatymo vietovėse (*rural*) skaičiavimai irgi atliekami pasitelkus vidutinio dydžio miestams skirtą koeficientą kartu naudojant sekančią išraišką:

$$C = -4,78(\log(f_c))^2 + 18,33 \log(f_c) - 40,98 \quad (5.16)$$

Skaičiuojant signalo stiprumą taške kur yra imtuvas, reikia susumuoti siųstuvo galią bei antenos stiprinimą ir atimti signalo perdavimo sistemos slopinimą kartu su kelio nuostoliais:

$$P_{RX} = P_{TX} + G_{TX} - L_{TX} - PL \quad (5.17)$$

čia: P_{RX} — priimamo signalo stiprumas (dBm), dBm; P_{TX} — siųstuvo galia (dBm); G_{TX} — siųstuvo antenos stiprinimas (dBd).

2.7. Išvados

Skyriuje aptarti Lietuvoje veikiančių mobiliojo ryšio technologijų dažniai, kanalų paskirstymas ir panaudojimas. Trumpai peržvelgta bazinių stočių veikimo bei jų dažnių planavimo metodika. Išnagrinėtos anteninių įrenginių savybės, apžvelgti pagrindiniai jų parametrai ir vertės.

Aptartas elektromagnetinio lauko energijos srauto tankis bei jo sąryšiai su elektrinio ir magnetinio lauko stipriais. Apžvelgti Lietuvoje energijos srauto tankį reglamentuojantys dokumentai bei Europos Sąjungos rekomendacija. Palyginus Europos Komisijos siūlomas vertes su Lietuvoje didžiausiomis leidžiamomis matyti, kad spinduliavimo intensyvumo vertinimas Lietuvoje yra iki 100 kartų griežtesnis.

Išnagrinėta Lietuvoje naudojama EML energijos srauto tankio skaičiavimo analitinė metodika, aptarti koeficientai, jų vertės. Taip pat peržvelgtas signalo stiprumo įvertinimo tolimajame lauke empirinis modelis.

3. EML VERTINIMO MODELIŲ REALIZACIJA

Šiame skyriuje aptariamas eksperimentinis elektromagnetinių laukų intensyvumo tyrimas ir gauti rezultatai. Pagal radiotechninio objekto parametrus atliekamas modeliavimas ir palyginami rezultatai bei įvertinamas tirtos elektromagnetinės situacijos adekvatumas. Matavimas buvo atliktas tiesioginio matavimo sąlygomis atviraime lauke. Panaudojus EMR–300 [20] elektromagnetinio lauko matuoklį buvo išmatuotas artimojoje antenos aplinkoje esantis EML energijos srauto tankis.

3.1. Tyrime naudota antena

„Banginio kanalo“, dar vadinama „Yagi“ antena — tai elementų rinkinys: aktyvus — vibratorius, pasyvūs — reflektorius ir direktoriai. Visi šie elementai tvirtinami tam tikra tvarka ir atstumais ant vieno skersinio laikiklio, kuris visada būna lygiagretus geometrinei antenos ašiai.

Tam tikro ilgio vibratorius sukuria elektromagnetines bangas, kurios tarsi perspinduliuojamos tolimesnių direktorių ir sukuria antrinę elektromagnetinę lauką. Nuo elemento ilgio priklauso jo rezonansinis dažnis, o nuo jo vietos — suderinimas. Norint anteną gerai suderinti su fideriu, reikia, kad jos varža būtų tik aktyvioji.

Yra įvairių banginio kanalo antenų, kuriose įgyvendinti: mažiausias antenos jautrumas pašaliniesiems signalams, pastovus stiprinimo koeficientas visoje dažnių juostoje, minimalūs šoniniai lapeliai antenos krypties diagramoje.

Pastaroji antena stipriausią elektromagnetinę lauką sukuria pagrindinio lapelio kryptimi. Jos turi būti orientuojamos taip, kad siuntimo kryptis būtų nukreipta į numatomą priimančios antenos vietą.

Tyrimui buvo naudojama Martin Steyer (DK7ZB) suprojektuota banginio kanalo, 19 elementų antena.

Jos parametrai:

Rezonansinis dažnis — 432 MHz;

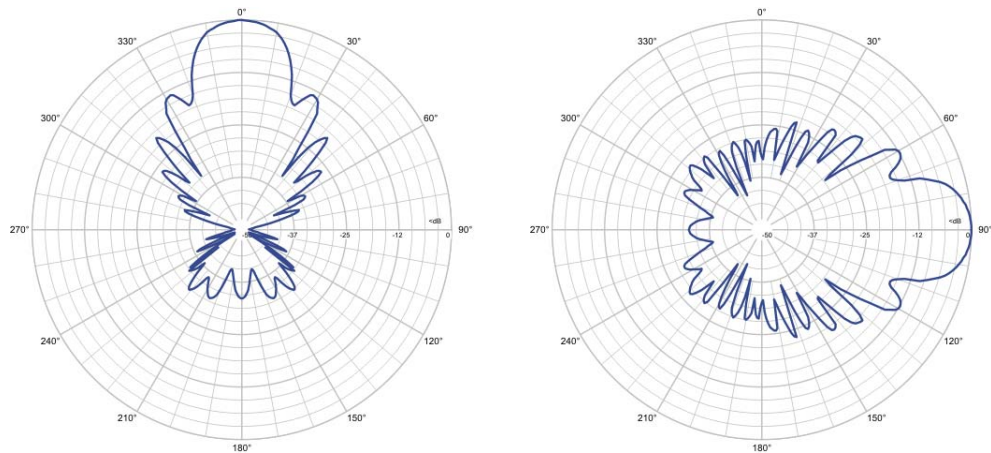
Stovinčios bangos koeficientas — $< 1,4:1$;

Antenos stiprinimas — 16,16 dBd;

Horizontalus kampas — 22° ;

Vertikalus kampas — 24° .

Pagal antenos kryptingumo diagramą buvo sugeneruotas informacinis failas, tinkantis daugumai padengiamumo ir spinduliavimo intensyvumo modeliavimo programų (3.1 pav.).



3.1 pav. 432 MHz, 19 elementų antenos horizontali (kairėje) ir vertikali (dešinėje) kryptingumo diagrama

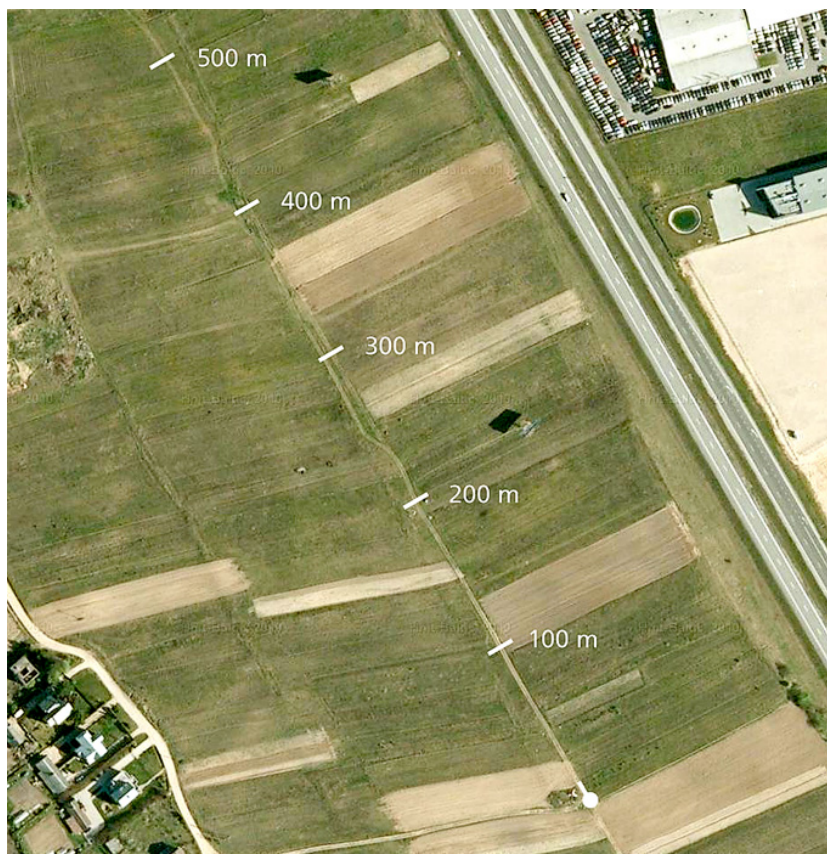
Dėl kryptingumo savybių, antenos sukuriama EML energijos srauto tankį reikės matuoti intensyviausia antenos spinduliavimo kryptimi. Eksperimentinio matavimo metu tikimasi išmatuoti ir šalutinius antenos kryptingumo charakteristikos lapelius.

3.2. Tyrimas, eiga ir gauti rezultatai

Matavimui pasirinktas Vilniaus rajone esantis plynas laukas, kurio ilgis didesnis kaip 1000 m, plotis daugiau kaip 300 m. Lauką iš pietų pusės supa vienos eilės medžių masyvas už kurių vėl laukas. Iš šiaurės pusės supa miškas, iš vakarų pusės — nedidelis medžių masyvas, už kurio privačių namų kvartalas, iš rytų pusės — automobilių kelias.

Kryptinė siuntimo antena buvo įrengta 500 m atstumu nuo pietinės lauko pusės ir atsukta į šiaurės pusę. Intensyviausia antenos spinduliavimo kryptimi vienoje tiesėje buvo sužymėti atskaitos taškai, kuriuose reikėjo išmatuoti EML energijos srauto tankį. Atstumai buvo parenkami tokia tvarka: 1, 2, 4, 6, 8, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 180, 200, 220, 240, 260, 280, 300, 350, 400. Kadangi didžiausias intensyvumas susidaro prie pat antenos, todėl norint tiksliai nustatyti EML energijos srauto tankį iš pradžių atstumai parenkami mažesniu žingsniu, o virš 150 m didinami. Kiekvienas taškas buvo išmatuotas 50 m ruletės pagalba ir kiekviename buvo tiesioginis matavimas su siuntimo antena.

Kiekviename tašką atliekamas 20 s matavimas. Buvo naudojamas EMR–300 prietaisas, turintis galimybę išmatuoti vidutinį intensyvumą per laiko vienetą. Matavimas buvo atliktas 34 taškuose, kuriuose matavimo aukštis (1,6 m) buvo fiksuotas stovo pagalba. Atliekant matavimus matomumas tarp prietaiso ir siųstuvo antenos nebuvo užstotas, taip sudarant kuo mažesnę įtaką tikslesniems rezultatams.



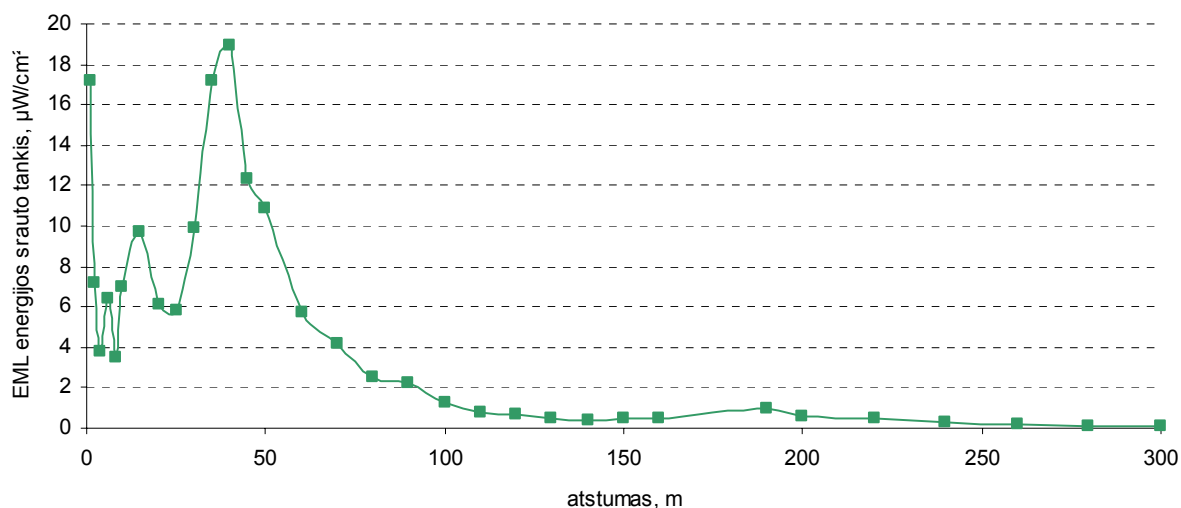
3.2 pav. Tyrimui pasirinktas laukas [21]

Siųstuvo ir antenos parametrai:

- antenos aukštis — 4,6 m;
- siųstuvo dažnis — 432 MHz;
- siųstuvo galia — 49 W;
- fiderio slopinimas — 1,303 dB;
- slopinimas dėl SBK — 0,081 dB;
- galia antenos įėjime — 38,5 W;

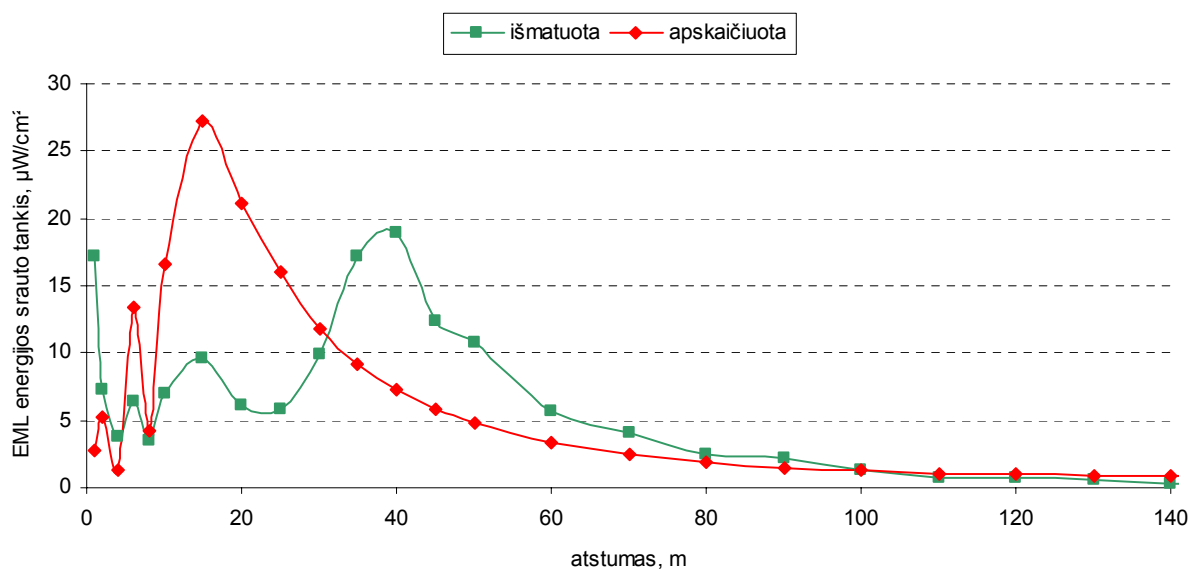
Elektromagnetinio lauko intensyvumo matavimo aktas pateikiamas B priede.

Iš 3.3 pav. matyti, kad pagrindinio antenos kryptingumo charakteristikos lapelio sudaroma zona prasideda ties 25 m, o nuo 30 m pradedamos viršyti higienos normose nustatytos vertės. 40 m atstumu nuo antenos išmatuotas intensyvumas siekė $18,9 \mu\text{W}/\text{cm}^2$. Panašaus dydžio intensyvumas buvo išmatuotas ir metro atstumu nuo antenos ($17,2 \mu\text{W}/\text{cm}^2$). Už 15 m nuo antenos išryškėja didžiausias šalutinis antenos kryptingumo charakteristikos lapelis, tačiau nustatytos vertės neviršijamos.



3.3 pav. EML energijos srauto tankio pasiskirstymas 1,6 m aukštyje

Buvo realizuotas EML stipriui apskaičiuoti skirtas modelis, kuriuo naudojantis gauti rezultatai buvo sulygininti su išmatuotais.



3.4 pav. Apskaičiuotas EML energijos srauto tankis 1,6 m aukštyje

Nors radiotechninio objekto parametrai buvo tiksliai žinomi, tačiau rezultatai nebuvo adekvatūs. Iš 3.4 pav. matyti, kad sumodeliuota didžiausia EML energijos srauto tankio vertė yra $27,2 \mu\text{W}/\text{cm}^2$, šią zoną, kurios maksimumas susidaro ties 15 m atstumu nuo antenos, sukuria pagrindinis antenos kryptingumo charakteristikos lapelis. Pirmojo šalutinio antenos kryptingumo charakteristikos lapelio sudaromos zonos maksimumas matomas ties 6 m ir yra $13,3 \mu\text{W}/\text{cm}^2$. Abejos kreivės pradeda koreliuoti didesniu kaip 100 m atstumu ten, kur pasireiškia didesnis laisvos erdvės slopinimas. Mažesniu atstumu kreivės tarpusavyje nekoreliuoja, nors tiek vaizduojančioje išmatuotas vertes, tiek ir apskaičiuotas vertes galima išskirti du šalutinio antenos kryptingumo charakteristikos lapelius bei vieną pagrindinį antenos kryptingumo charakteristikos lapelį.

3.3. Išvados

Pateiktas eksperimentinis elektromagnetinių laukų intensyvumo tyrimas ir gauti jo rezultatai. Pagal radiotechninio objekto parametrus buvo atliktas modeliavimas ir įvertintas tirtos elektromagnetinės situacijos adekvatumas. Matavimas buvo atliktas su atestuotu matavimo prietaisu EMR–300 pagal Lietuvos higienos normose apibrėžtą elektromagnetinio lauko įvertinimo metodiką. Artimajame antenos lauke buvo išmatuotas EML energijos srauto tankis, kuris atstume nuo 30 iki 50 metrų viršijo leistinas higienos normas.

Iki 100 m atstumu išmatuotos ir apskaičiuotos vertės tarpusavyje nekoreliuoja, nors galima išskirti vienodai išsidėsčiusius pagrindinį ir šalutinius antenos kryptingumo charakteristikos lapelius. Tokia neadekvati situacija galėjo susidaryti dėl antenos gamintojo pateiktos klaidingos kryptingumo charakteristikos ar antenos sumontavimo kokybės.

4. EML ĮVERTINIMO PROGRAMOS KŪRIMAS

Šiame skyriuje aptariamas loginis sukurtos elektromagnetinio spinduliavimo intensyvumo įvertinimo programos veikimas, patikrinama ar programos gauti apskaičiuoti rezultatai yra teisingi.

4.1. Programos veikimo aprašymas

Programa visus nustatymus išsaugo konfigūracijos faile, todėl startuojant programą užkraunami anksčiau naudoti spalvų, koordinatų, pritraukimo parametrai. Norint į radiotechninių objektų masyvą užkrauti anksčiau išsaugotus objektus, reikia pasirinkti norimą duomenų failą.

Programoje esantys žemėlapiai vaizduojami pagal tris pagrindinius parametrus — žemėlapių sluoksnį, pritraukimą ir koordinates. Žemėlapių koordinatų sistema — metrinė LKS-94. Kiekvienas žemėlapių pritraukimo lygis reiškia vis kitą metro ir pikselio ekrane santykį. Kuo žemėlapių pritraukimo lygis didesnis, tuo santykis mažesnis ir gali siekti net 0,5 m viename pikselyje. Programoje žemėlapyje vaizduojamų taškų tikslumas yra ± 1 m. Pasiiekti didesniam tikslumui, metrines koordinates reikėjo apdoroti ne kaip sveikuosius skaičius, o kaip skaičių su slankiuoju kableliu.

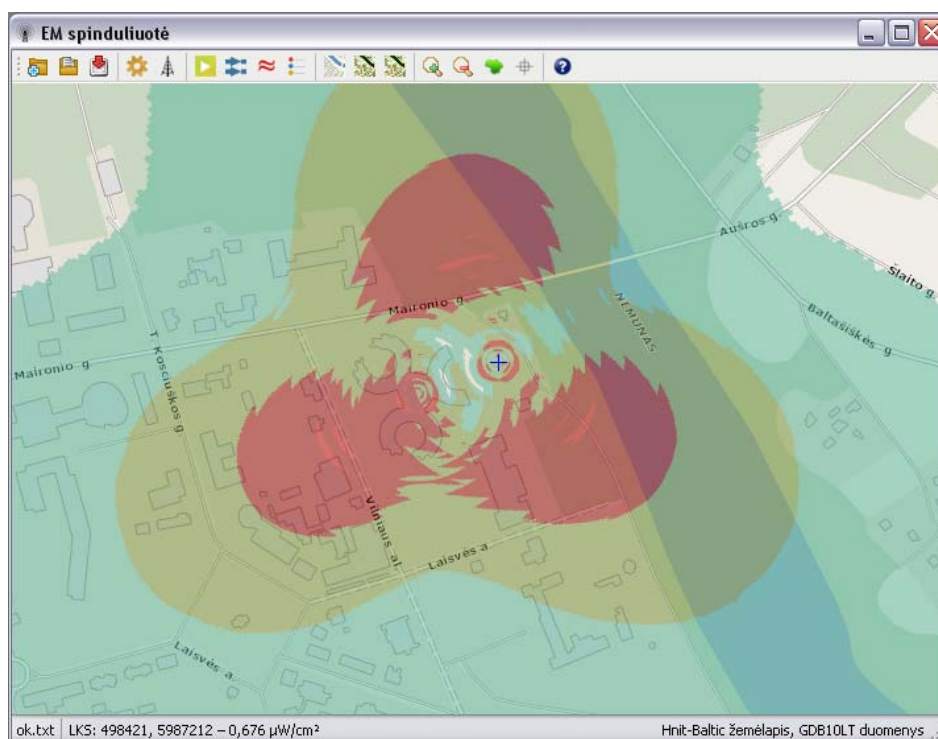
Žemėlapių paveikslėlis sudarytas iš daug mažo dydžio paveikslėlių, kurie patalpinti internetiniame serveryje. Žinant norimos ekrane vaizduoti vietovės koordinates, reikia atlikti skaičiavimus tam, kad būtų gauti mažųjų paveikslėlių adresai. Pagal apskaičiuotus adresus paveikslėliai yra parsiuoiami į kompiuterį ir paeiliui sudėliojant suformuojamas vientisas žemėlapis. Kiekvienas žemėlapių pastūmimas, tai naujos centro koordinatės parinkimas ir mažųjų paveikslėlių įkėlimo uždavinys.

Iš duomenų failo užkrauti radiotechninių objektų duomenys įrašomi į masyvą. Pastarajame esančiais duomenimis galima lengvai manipuluoti. Radiotechninių objektų parametrų lange kaip tik ir galima pakeisti šio masyvo kintamuosius.

Atliekant modeliavimo uždavinį, naudojamas aukščiau paminėtas analitinis elektromagnetinių laukų intensyvumo įvertinimo artimoje bazinių stočių aplinkoje metodas. Visi duomenų masyve esantys radiotechninių objektų duomenys yra panaudojami modeliavime. Modeliavimas vykdomas kiekvienam radiotechniniam objektui atskirai. Pirmiausiai yra įvertinamas kiekvieno objekto sukuriamas elektrinio lauko dydis. Pastarasis dydis nekinta viso objekto modeliavimo metu. Sekantis uždavinys — įvertinti objekto elektrinio lauko dydį skirtinguose taškuose, šiuo atveju Dekarto koordinatų sistemoje. Tai pasikartojantis uždavinys, kuris kiekviename metre įvertina kryptingumo

diagramos dydį, atstumą nuo antenos ir apskaičiuoja galutinį objekto sukuriamą elektromagnetinio lauko intensyvumą.

Kiekvienam objektui modeliavimo metu įvertinamas elektromagnetinis lauko intensyvumas pasirinktu spinduliu nuo antenos. Jeigu skaičiuojamas 300 m spindulys, duomenų atžvilgiu tai sudaro masivą iš 600 stulpelių ir 600 eilučių. Toks masivas įrašomas į radiotechninio objekto modeliavimo duomenų failą su pradžios koordinatėmis. Įvertinant suminį elektromagnetinio lauko intensyvumą, reikia nuskaityti duomenų failus su pradžios koordinatėmis, ir susumuoti laukelius, kurie turi vienodas koordinatas. Taip gaunamas suminio elektromagnetinio lauko intensyvumo duomenų masivas, kurio centre yra tiriamas radiotechninis objektas.



4.1. pav. Sukurtos programos pagrindinis langas

Gauti galutiniai duomenys vizualizuodami žemėlapyje (4.1 pav.). Pagal suminį duomenų masivą nubraižomas paveikslėlis, kuriame kelios skirtingos spalvos žymi skirtingo elektromagnetinio lauko intensyvumo zonas. Pastarasis paveikslėlis pagal centrinio radiotechninio objekto koordinatas perkliamas ant žemėlapijo taip suteikiant galimybę vizualiai įvertinti laukų intensyvumą. Esant poreikiui gautus rezultatus analizuoti polinėje koordinatinių sistemoje, programoje yra įdiegtas įrankis, kuris iš suminio duomenų masivo kas 5° ir nustatytais atstumais išrenka elektromagnetinio lauko intensyvumo vertes ir atvaizduoja jas lentelėje. Duomenis galima eksportuoti ir analizuoti kitose programose.

Programos vartotojo patogumui yra įdiegtas atstumo matavimo įrankis, kuris matuoja atstumą ekrane tarp dviejų pažymėtų taškų. Gautas atstumas pikseliais, atsižvelgiant į žemėlapiu pritraukimo mastelį yra perskaičiuojamas į metrus. Esant poreikiui programoje galima operuoti metrinėje koordinatinių sistemoje, bei pasaulinėje geodezinių koordinatinių sistemoje. Programa uždavinius susijusius su koordinatėmis, atlieka metrinėje sistemoje, tačiau jeigu bus pasirinktas geodezinių koordinatinių atvaizdavimas — pastarosios bus tiesiog perskaičiuojamos.

Nagrinėjant elektromagnetinius laukus patogiu naudoti įrankį, kuris atlieka pasirinktos antenos parametrų keitimą neperskaičiuojant visų likusių sistemoje. Tokiu būdu sutaupomas vartotojo laikas bei galima vaizdžiai pamatyti, kurie parametrai įtakoja draudžiamųjų zonų kitimą. Skaičiavimai atliekami su vieno objekto duomenimis — po kiekvieno naujo parametro įvedimo tiriamo ir likusių objektų duomenų masyvai tiesiog perklojami iš naujo.

Spinduliavimo intensyvumui sumažinti programoje sukurtas įrankis leidžiantis pasirinktuose taškuose atlikti vieno ar kelių radiotechninių objektų spinduliavimo įvertinimą. Programai reikia nurodyti, kokius parametrus ir kokiame reikšmių diapazone leidžiama keisti. Geriausios vertės ieškomos nuosekliai, tikrinant kiekvieną parametą su kiekvienu. Atlikus skaičiavimus programa pateikia geriausią surastą rezultatą. Taip pat vertinama kaip pasikeitė signalo stiprumas tolimajame lauke. Skaičiavimai atliekami pagal *Okumura-Hata* modelį, dažnį taikant pagal nurodytą antenos duomenų failę. Programa visus skaičiavimo rezultatus išsaugo į tekstinį failą, kurį galima analizuoti su kitomis programomis.

Programos naudojimo aprašymas pateikiamas A priede.

4.2. Programos skaičiavimų patikrinimas

Pasirinkus vienodus tiriamų objektų parametrus ir atlikus rankinius skaičiavimus bei automatinis įvertinimus paaiškėjo, kad programa skaičiavimus atlieka tiksliai. Patikimumui ištirti buvo pasirinkta „Kathrein 80010292“ antenos kryptingumo diagrama. Skaičiavimai buvo atlikti intensyviausia spinduliavimo kryptimi, kai antena iškelta į 20 m aukštį. Tiriamas taškas buvo 1,5 m aukštyje ir judėjo tolyn nuo antenos 300 m. Gauti rezultatai parodė, jog esant labai mažoms vertėms vidutiniškai 1 % skirtumas atsirado dėl apvalinimo sąlygų skirtumo.

4.3. Išvados

Programa sukurta C# programavimo kalba, žemėlapiai naudojami iš laisvai internete prieinamų šaltinių. Pagal darbinės koordinatės programa parsisiunčia žemėlapių dalis iš internetinio serverio ir suformuoja vientisą vaizdą. Modeliavimui naudojamas analitinis elektromagnetinių

laukų intensyvumo įvertinimo artimoje bazinių stočių aplinkoje metodas. Gaunami rezultatų masyvai vieno metro tikslumu pasirinktu atstumu nuo tiriamo radiotechninio objekto. Programa atlikus skaičiavimus gali automatiškai įvertinti geriausią parametrų rinkinį nurodytai antenai kai yra pažymėtos svarbios zonos. Sukurta galimybė vaizdžiai keisti pasirinkto objekto parametrus kas kartą neperskaičiuojant visų rodomų radiotechninių objektų spinduliavimo intensyvumo. Atliktas programos skaičiavimo patikimumo patikrinimas parodė, jog tik esant ypač mažiems skaičiams programos vertės skyrėsi vidutiniškai 1 % nuo apskaičiuotų rankiniu būdu. Šis skirtumas atsirado dėl griežtesnio apvalinimo taisyklių sukurtoje programoje.

5. EML MINIMIZAVIMO GALIMYBIŲ TYRIMAS

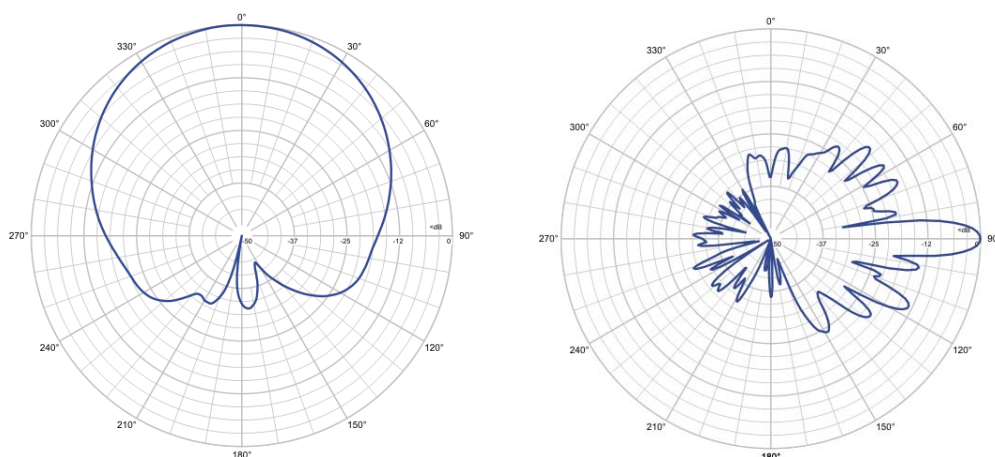
Šiame skyriuje išnagrinėta, kaip skirtingi antenų parametrai lemia EML energijos srauto tankį. Modeliavimams naudojama sukurta programa, kuri skaičiavimus atlieka pagal Lietuvoje dažniausiai naudojamą analitinį modelį. Taip pat nagrinėjama, kokią įtaką spinduliavimo intensyvumui daro kompleksiskai keičiami antenų parametrai. Pateikiami EML energijos srauto tankio sumažinimo galimybių rezultatai bei aptariama, kaip pasikeičia paslaugos kokybė vertinant tolimąjį lauką.

Pagrindiniai parametrai, kuriuos parenkant galima sumažinti EML energijos srauto tankį:

- Siųstuvo galingumas;
- Antenos palenkimas;
- Antenos kryptis;
- Antenos iškėlimo aukštis;
- Antenos įrengimo vieta.

5.1. Energijos srauto tankio modeliavimo parametrai

Visi modeliavimai šiame skyriuje buvo atlikti pasinaudojant sukurta taikomąja programa EML energijos srauto tankiui vertinti. Skaičiuojama buvo 1,5 m aukštyje, jeigu nenurodyta kitaip, antena pakelta į 20 m aukštį. Siųstuvo galia 20 W, signalo perdavimo linija nuostolių neturi. Antena mechaniškai bei elektriškai nepalenkta, skaičiavimai vykdyti intensyviausia spinduliavimo kryptimi, jeigu nenurodyta kitaip.

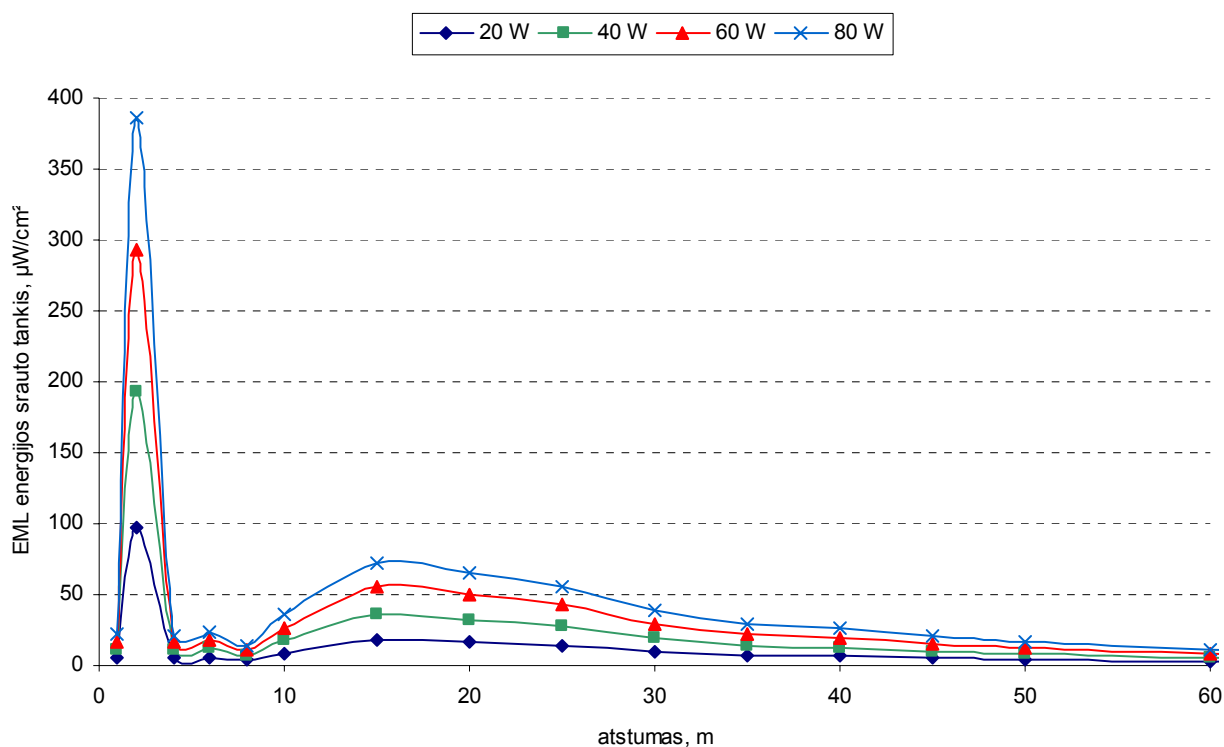


5.1 pav. „Kathrein 80010292“ antenos horizontali (kairėje) ir vertikalė (dešinėje) kryptingumo diagrama

Panaudota „Kathrein 80010292“ antenos normuota kryptingumo diagrama (5.1 pav.). Antenos rezonansinis dažnis — 1855 MHz, stiprinimas — 14,12 dBd, elektrinis palenkimas — 0°, poliarizacija — -45°, horizontalus spinduliavimo kampas — 68,62°, vertikalus spinduliavimo kampas — 8,06°.

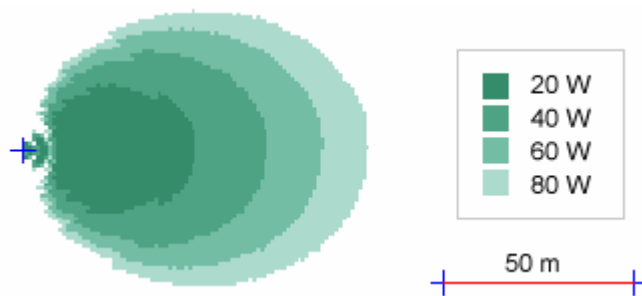
5.2. Antenos išspinduliuojamos galios įtaka energijos srauto tankiui

5.2 pav. matyti kaip kinta EML energijos srauto tankis priklausomai nuo siųstuvo galios. Skaičiavimai atlikti 19 m aukštyje esant keturioms skirtingoms galioms. Paprastai viename sektoriuje įrengiamai technologijai įdiegiama nuo 1 iki 4 kanalų. Kiekvienas kanalas sukuria 20 W galią. Iš kreivių matyti, jog 2 m atstumu nuo antenos EML energijos srauto tankis viršijamas tiek esant vienam, tiek ir keturiems kanalams. Esant vienam kanalui (20 W) sukuriamas intensyvumas lygus 97,2 $\mu\text{W}/\text{cm}^2$. Grafike matyti (5.2 pav.), jog EML energijos srauto tankio priklausomybė nuo siųstuvo galios yra tiesinė tiek šalutinio antenos kryptingumo charakteristikos lapelio zonoje, tiek ir pagrindinio antenos kryptingumo charakteristikos lapelio zonoje.



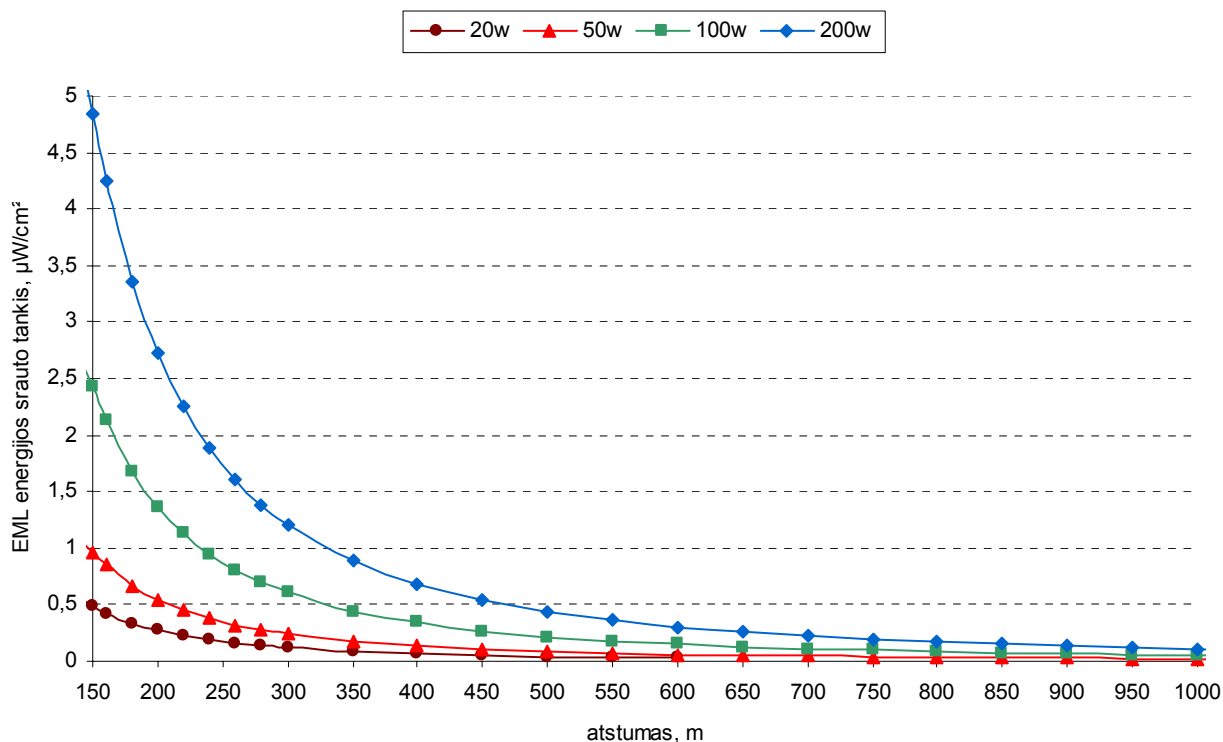
5.2 pav. EML energijos srauto tankis esant skirtingai siųstuvo galiai 19 m aukštyje

5 $\mu\text{W}/\text{cm}^2$ spinduliavimo intensyvumą pagal plotą, priklausomai nuo siųstuvo galios matome 5.3 pav., čia antena 20 m aukštyje, skaičiavimai atliekami 19 m aukštyje.



5.3 pav. Intensyvumas esant skirtingai siųstuvo galiai 19 m aukštyje, kai riba $5 \mu\text{W}/\text{cm}^2$

Iš 5.4 pav. esančio grafiko matyti, jog didelės suminės galios siųstuvas (200 W), esant didesniai kaip 500 m atstumui nuo antenos, nesukuria didesnio kaip $0,5 \mu\text{W}/\text{cm}^2$ EML energijos srauto tankio. Įprastai nagrinėjamo 20 W galios siųstuvo sukuriamas intensyvumas 300 m atstumu yra tik $0,12 \mu\text{W}/\text{cm}^2$.

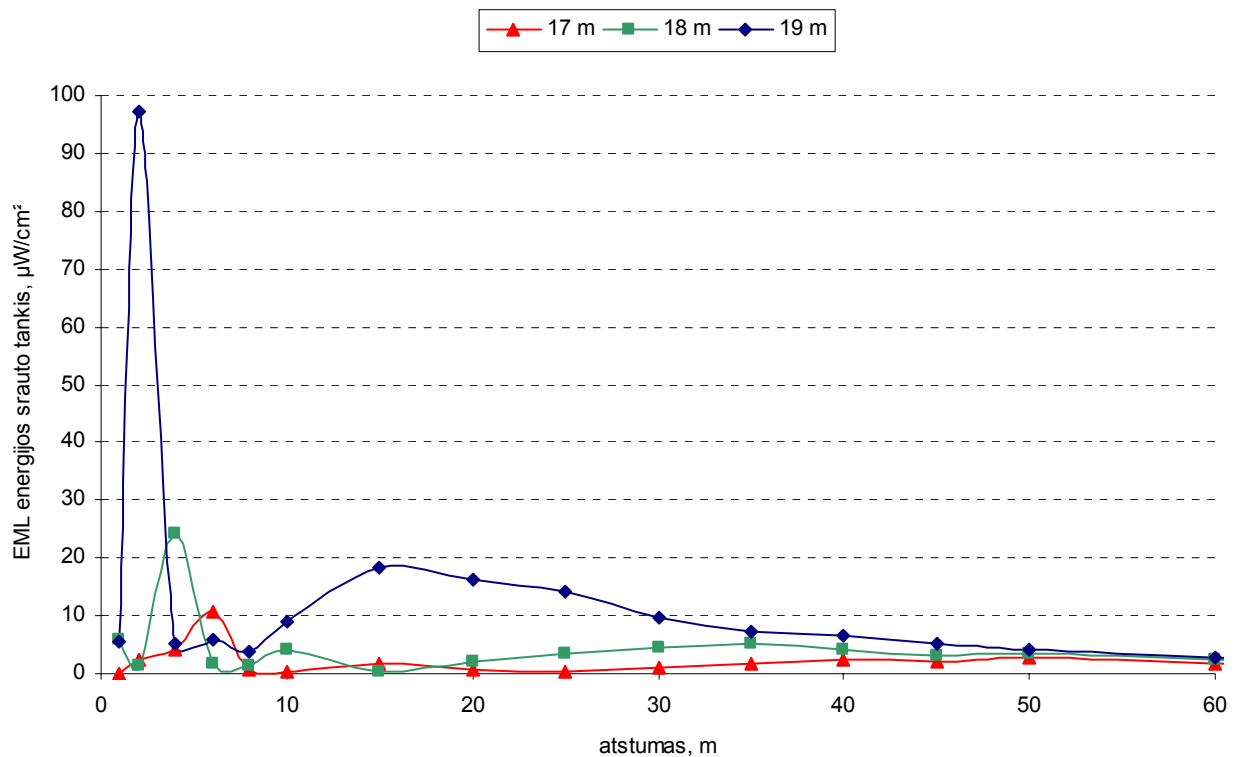


5.4 pav. EML energijos srauto tankis esant skirtingai siųstuvo galiai 19 m aukštyje

Todėl, skaičiuojant suminį EML energijos srauto tankį, reikėtų atsižvelgti į radiotechninius objektus, esančius ne toliau kaip 300 m nuo tiriamojo taško. Jeigu radiotechninis objektas dirba didesne galia (200 W), tokį objektą reikėtų įtraukti į tyrimą ir kai atstumas yra 500 m nuo tiriamojo taško.

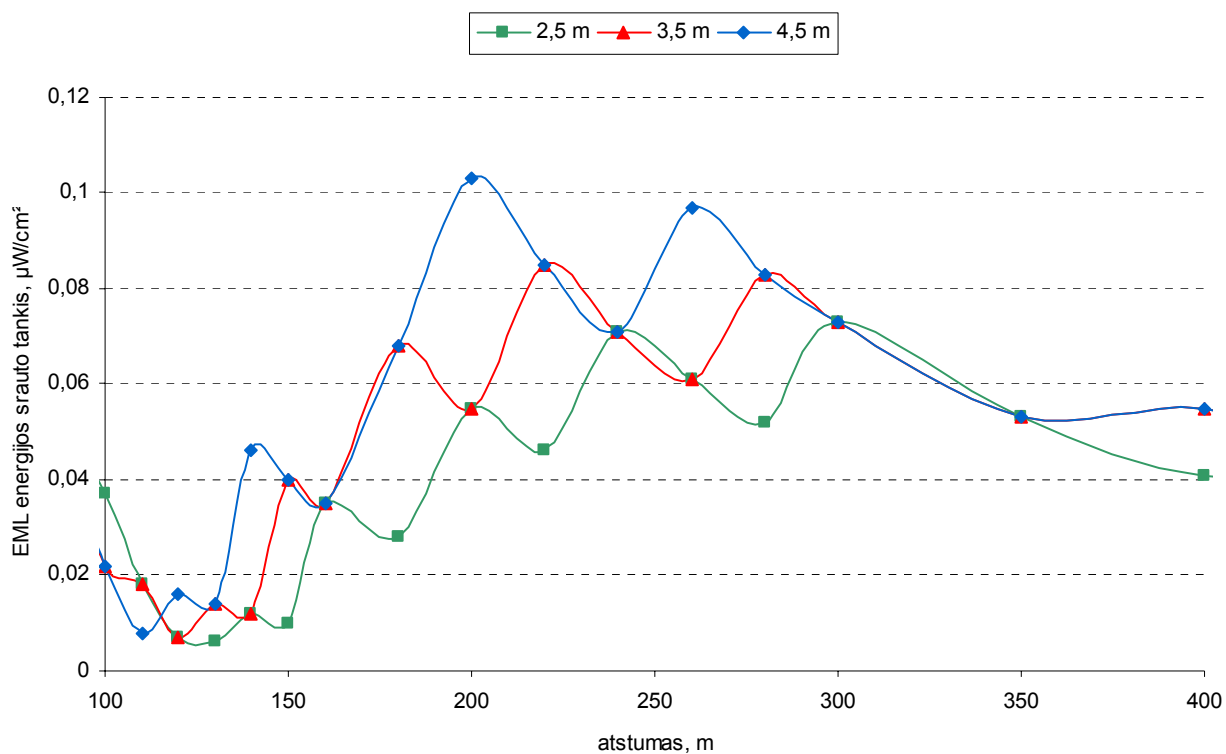
5.3. Antenos bei matavimo aukščio įtaka energijos srauto tankiui

Pateiktame grafike (5.5 pav.) vaizduojami EML energijos srauto tankio modeliavimo rezultatai, kai yra keičiamas skaičiavimo aukštis.



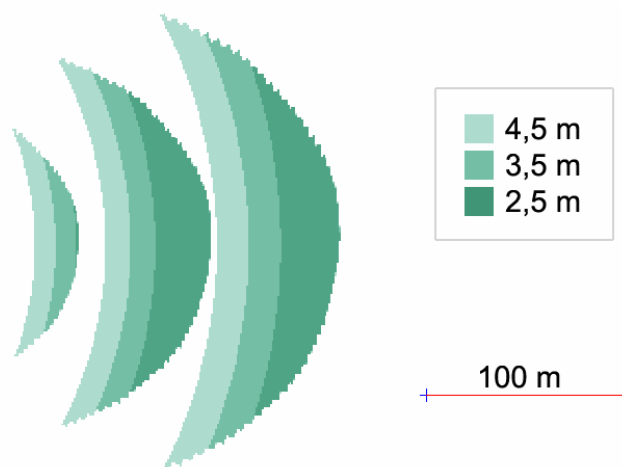
5.5 pav. Skirtinguose aukščiuose sumodeliuotas EML energijos srauto tankis

Galime pastebėti jog 19 m aukštyje, kai antena iškelta į 20 m aukštį, už 2 m, susidaro ypač didelis $97,09 \mu\text{W}/\text{cm}^2$ intensyvumas, kuris $87,09 \mu\text{W}/\text{cm}^2$ viršija higienos normas. Skaičiavimus atlikus vienu metru žemiau, tokiu pačiu atstumu EML energijos srauto tankis tesiekia $1,39 \mu\text{W}/\text{cm}^2$. Stebime šalutinio antenos kryptingumo charakteristikos lapelio persistūmimą tolyn nuo antenos. 19 m aukštyje ties 2 m, 18 m aukštyje ties 4 m, o 17 m aukštyje ties 6 m. Reikia pastebėti, kad lapelis ne tik stumiasi tolyn, bet ir mažėja jo maksimaliai sukuriamas intensyvumas — 19 m aukštyje $97,09 \mu\text{W}/\text{cm}^2$, 18 m aukštyje $24,27 \mu\text{W}/\text{cm}^2$, 17 m aukštyje $10,79 \mu\text{W}/\text{cm}^2$. Pagrindinis antenos kryptingumo charakteristikos lapelis 19 m aukštyje taip pat viršija higienos normas $8,28 \mu\text{W}/\text{cm}^2$. Atlikus skaičiavimus metru žemiau lapelio spinduliuojama zona pasistūmusi pirmyn ir dėl slopinimo laisvoje erdvėje higienos norma jau nėra viršijama.



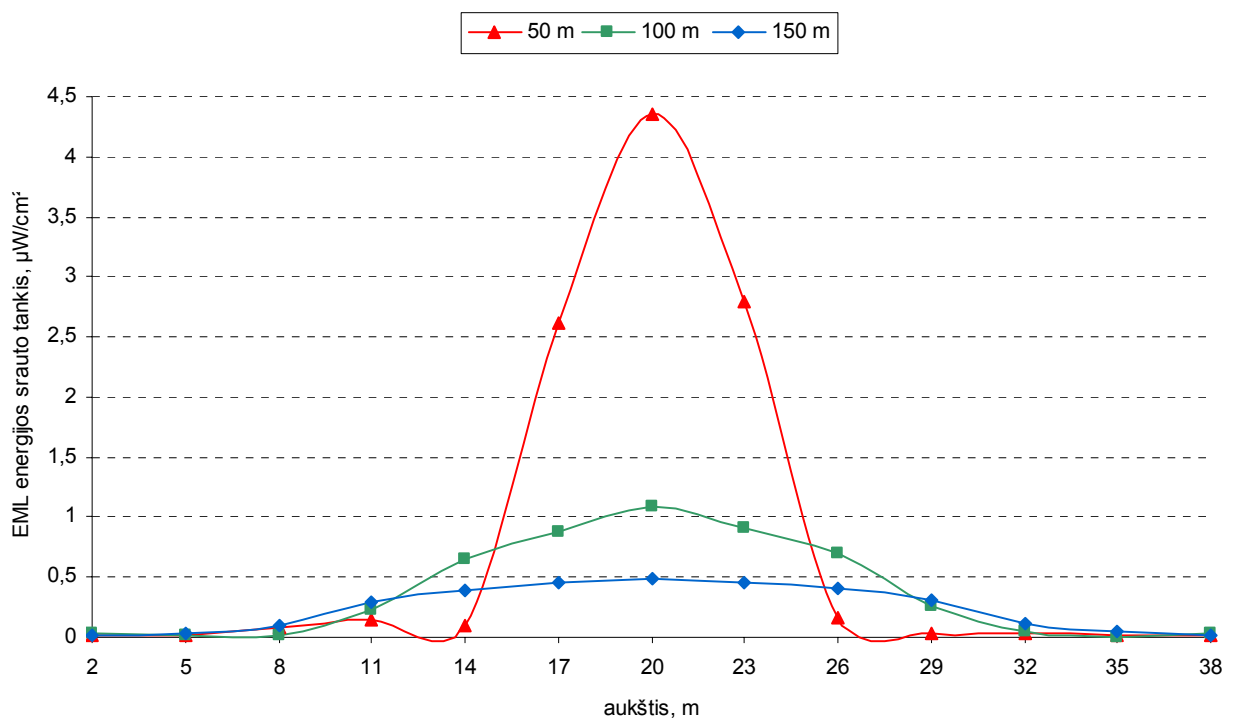
5.6 pav. Skirtinguose aukščiuose sumodeliuotas EML energijos srauto tankis

5.6 pav. nubraižytas grafikas pagal EML energijos srauto tankio modeliavimo rezultatus, kai pasirinktas daug mažesnis skaičiavimo aukštis negu buvo aptariamas 5.5 pav. Matome, jog didinant matavimo aukštį susidarančių intensyvesnių zonų atstumas nuo antenos tiesiškai mažėja. Keičiasi ir intensyvumo vertė. Modeliuojant 2,5 m aukštyje didžiausia vertė buvo $0,073 \mu\text{W}/\text{cm}^2$ ties 240 m atskaita. Pakilus 1 m aukštyn, intensyvumo pikas priartėjo 20 m ir padidėjo iki $0,085 \mu\text{W}/\text{cm}^2$. EML energijos srauto tankis lygus $0,103 \mu\text{W}/\text{cm}^2$ buvo išmatuotas ties 200 m nuo antenos.



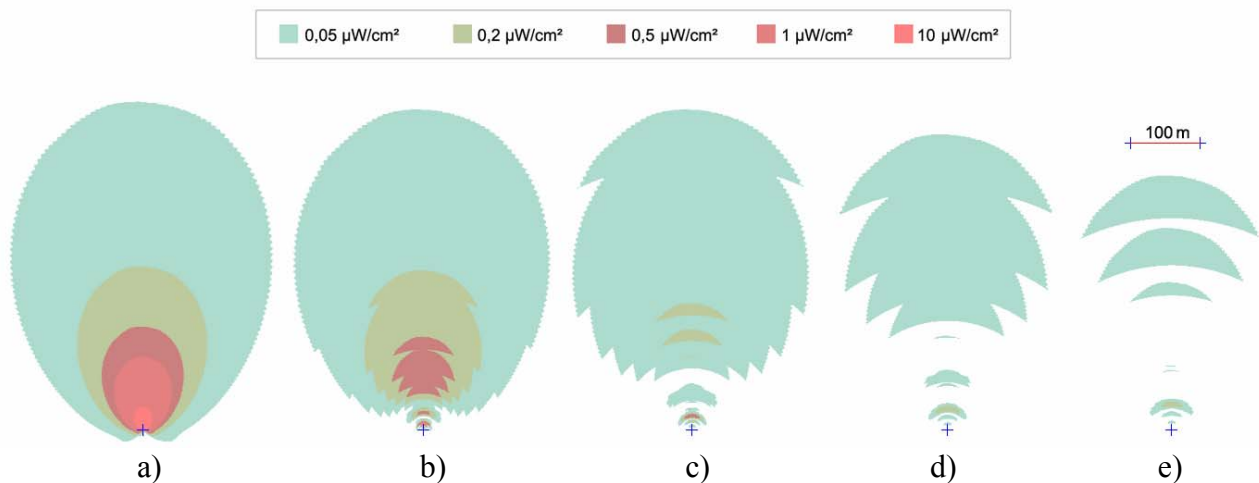
5.7 pav. Skirtingo dydžio, to paties intensyvumo ($0,066 \mu\text{W}/\text{cm}^2$) plotai, kai skiriasi aukštis

5.7 pav. pavaizduoti plotai, kuriuose EML energijos srauto tankis yra vienodas, tačiau apskaičiuotas skirtinguose aukščiuose. Plotai buvo modeliuojami nurodžius, jog vaizduojama riba yra $0,066 \mu\text{W}/\text{cm}^2$ ir mažiau. Vaizdžiai matome kaip keičiant skaičiavimo aukščius, antenos sukuriamas tokio pačio EML energijos srauto tankio plotas laipsniškai kinta — kuo aukščiau antenos, tuo tokio paties intensyvumo dengiamas plotas didesnis. Nors plotuose intensyvumas labai mažas, bet tokiu pačiu principu galima vadovautis skaičiuojant didelį intensyvumą. Kuo arčiau antena — tuo didesnis tos pačios vertės dengiamas plotas.



5.8 pav. Santykis tarp matavimo aukščio bei EML energijos srauto tankio
50 m, 100 m, 150 m atstumais

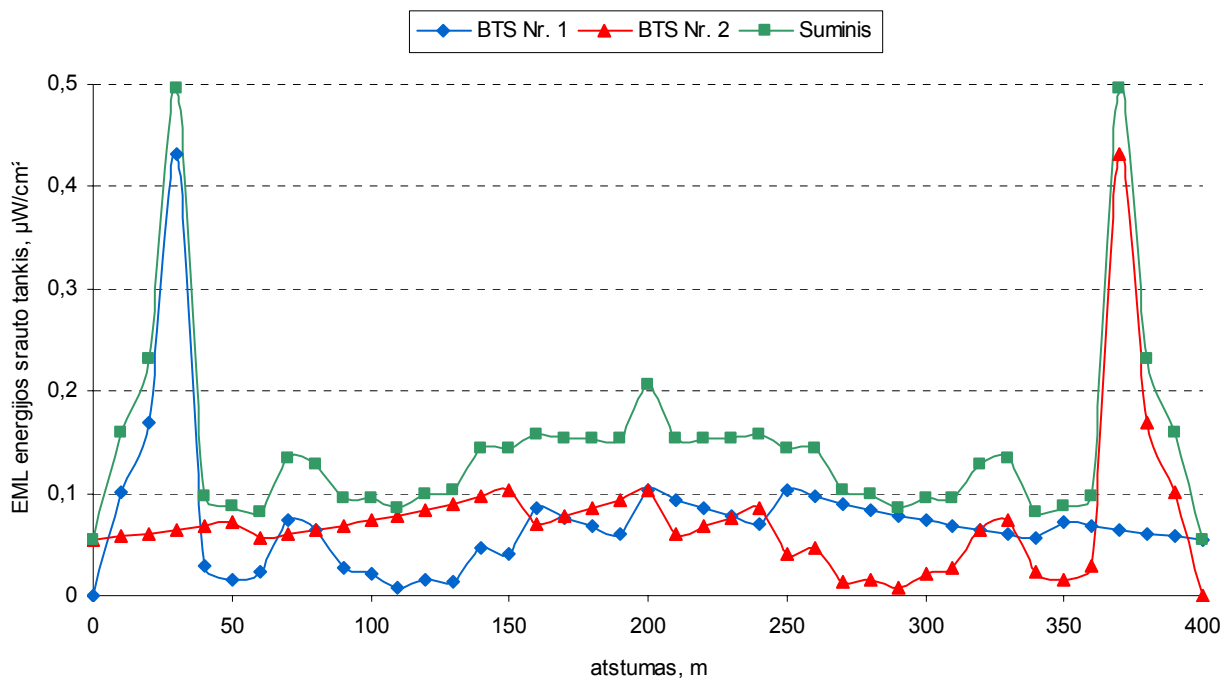
Brėžinyje pateikiamas santykis tarp matavimo aukščio bei EML energijos srauto tankio esant didesniems atstumams nuo spinduliuojančios antenos (5.8 pav.). Antena iškelta į 20 m aukštį, skaičiavimas vykdytas nuo 2 m iki 38 m. 50 m atstumu apskaičiuoto intensyvumo vertėms aiškia įtaką turėjo pagrindinis antenos kryptingumo charakteristikos lapelis ir jo kitimas. Skaičiuojant antenos aukštyje gautas EML energijos srauto tankis siekė $4,35 \mu\text{W}/\text{cm}^2$. Tuo tarpu 14 m aukštyje intensyvumas buvo beveik nepastebimas. Kreivė, žyminti skaičiavimą 150 m atstumu kinta tolygiai ir iš to galima spręsti, kad tokiu atstumu skaičiavimo aukštis intensyvumo vertes nulemia nežymiai.



5.9 pav. EML energijos srauto tankis kintant skaičiavimo aukščiui
a) 20 m, b) 15 m, c) 10 m, d) 5 m, e) 1,5 m

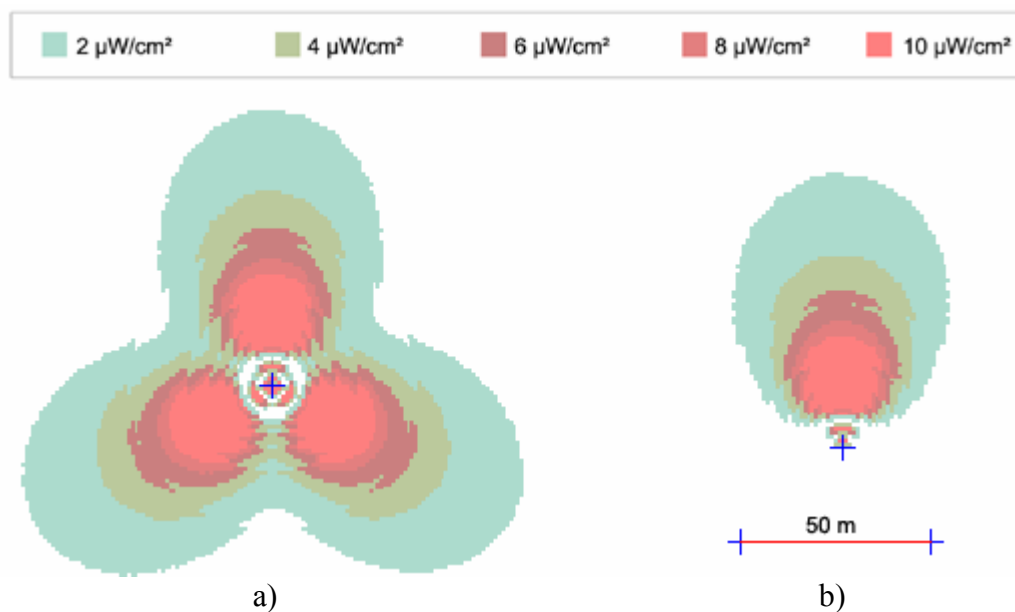
5.4. Gretimų antenų įtaka energijos srauto tankiui

Galima pastebėti, jog elektromagnetinio lauko energijos srauto tankis pasirinktame taške priklauso ne tik nuo artimiausio spinduliuojančio objekto (5.10 pav.). Apskaičiuojant intensyvumo vertes esant nedidelėms galioms, būtina atsižvelgti į visus 300 m atstumu išsidėsčiusius radiotechninius objektus. Todėl kiekviename paviršiaus taške sumuojamas visų pasirinktų spinduliuojančių objektų EML energijos srauto tankis.



5.10 pav. Dviejų bazinių stočių sukuriami EML energijos srauto tankiai ir jų suma

Grafike matomas dviejų vienodų antenų sukuriamas intensyvumas. Pirmosios bazinės stoties antena yra ties koordinatų atskaitos pradžia, antroji — už 400 metrų. Modeliavimo rezultatai gauti skaičiavimus atliekant 5 m aukštyje, kai antenos įrengtos 20 m aukštyje. Tolstant nuo antenos, intensyviausia spinduliavimo kryptimi, ryškiausias maksimumas matomas už 30 m — šalutinis antenos kryptingumo charakteristikos lapelis. Nuo 140 m pradeda ryškėti energijos srauto tankio padidėjimas, tai pagrindinis antenos kryptingumo charakteristikos lapelis. Grafike taip pat atsispindi suminis abiejų antenų sukuriamo elektromagnetinio lauko energijos srauto tankis. Matome, jog BTS1 intensyvumas už 350 m pradeda mažėti, tačiau sąveikaudamas su BTS2 šalutiniu antenos kryptingumo charakteristikos lapeliu bendrą elektromagnetinio lauko energijos srauto tankis vis tiek padidina nuo $0,43 \mu\text{W}/\text{cm}^2$ iki $0,5 \mu\text{W}/\text{cm}^2$ ties 380 m.

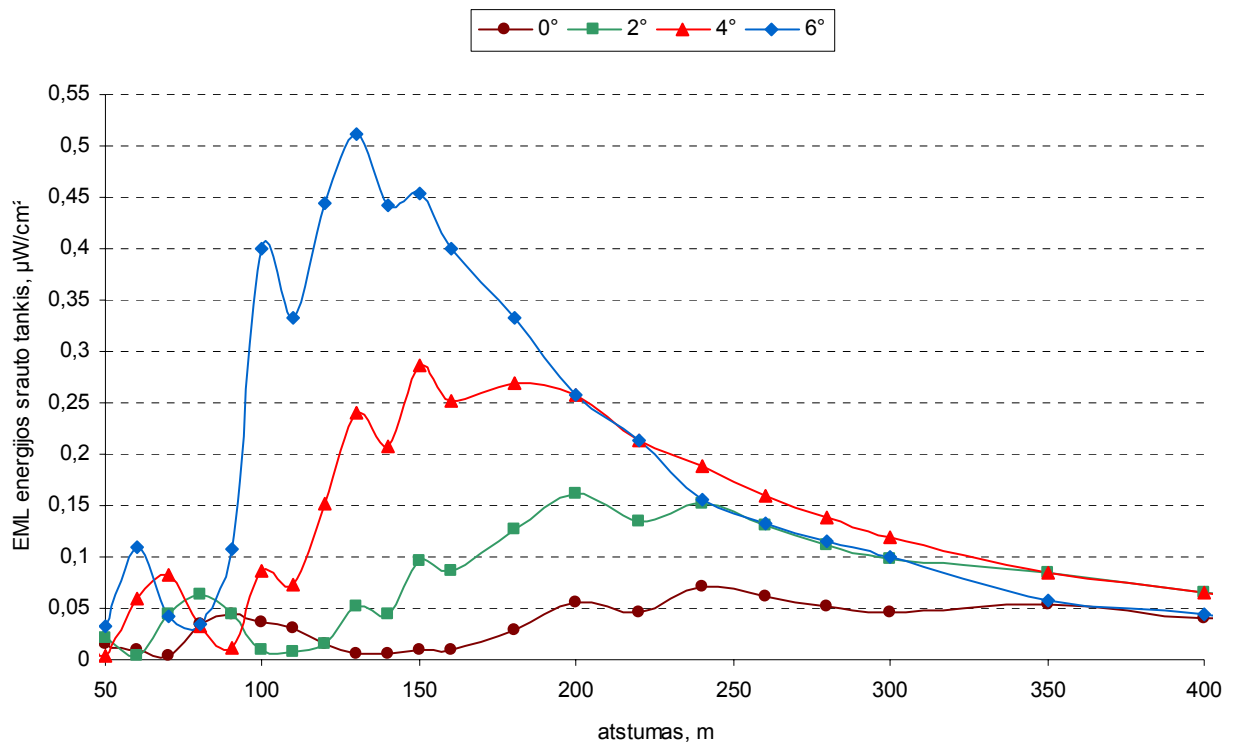


5.11 pav. EML energijos srauto tankio pasiskirstymas a) kelių antenų suminis, b) vienos antenos

Gretimų antenų įtaką vaizdžiai parodo 5.11 pav., kurio dešinėje pavaizduotas vienos antenos EML energijos srauto tankio pasiskirstymas. Kairėje — EML energijos srauto tankio pasiskirstymas, kai trys antenos sumontuojamos viename taške pasukant jas kas 120° . Matome, jog plotas, kuriame ties antena susikuria $2 \mu\text{W}/\text{cm}^2$ padidėja. Taip atsitinka dėl plačios antenos horizontalios spinduliavimo diagramos. Mažos šoninio spinduliavimo vertės esant vienai antenai nepastebimos, tačiau montuojant kelias antenas įtaką sudaro.

5.5. Antenų palenkimo ir pasukimo įtaka energijos srauto tankiui

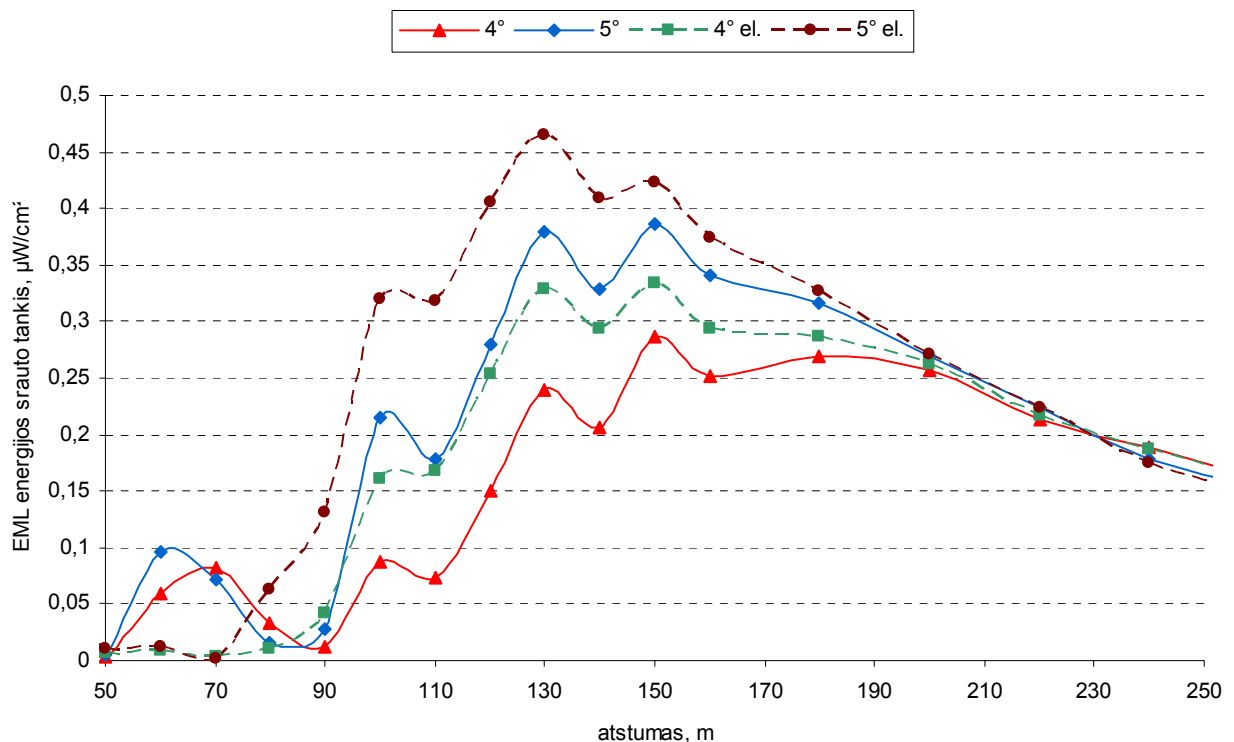
Antenos palenkimas — parametras, kuriuo galima reguliuoti antenos nuosvyrį arba palenkimą ir taip minimizuoti EML energijos srauto tankį. Esant nuosvyrio parametrai pagrindinio antenos kryptingumo charakteristikos lapelio sukuriama intensyvumo zoną galima perstumti toliau arba arčiau antenos atžvilgiu. Žinoma, jog tuo pačiu keičiasi ir šalutinių antenos kryptingumo charakteristikos lapelių sukuriamų intensyvumo zonų padėtys.



5.12 pav. EML energijos srauto tankis esant skirtingiems palenkimo kampams

Mechaninis palenkimas įgyvendinamas keičiant fizinę antenos padėtį paviršiaus atžvilgiu. 5.12 pav. pateiktame grafike matome, kaip kinta EML energijos srauto tankis esant skirtingiems palenkimo kampams, matavimus atliekant 1,5 m aukštyje. Kuo didesnis kampas, tuo arčiau pagrindinis antenos kryptingumo charakteristikos lapelis sukuria intensyvią zoną. Esant 0° palenkimui, už 240 m sukuriamas 0,07 $\mu\text{W}/\text{cm}^2$ energijos srauto tankis. Didėjant palenkimo kampui keičiasi ne tik veikiamas paviršiaus plotas, bet ir didėja intensyvumas jame. 2° palenkimas už 200 m sukuria 0,16 $\mu\text{W}/\text{cm}^2$ energijos srauto tankį. Dar dviem laipsniais daugiau palenkta antena sukuria 0,29 $\mu\text{W}/\text{cm}^2$ energijos srauto tankį už 150 m. Galiausiai 0,51 $\mu\text{W}/\text{cm}^2$ intensyvumas sukuriamas 130 m atstumu. Matome, jog EML energijos srauto tankis labai priklauso nuo

palenkimo kampo. Palenkimą sumažinus dviem laipsniais intensyviausioje spinduliavimo zonoje galime gauti net per pus mažesnę energijos srauto tankį.



5.13 pav. EML energijos srauto tankis kai antena palenkta mechaniškai ir elektriškai

Elektrinis nuosvyris kontroliuojamas keičiant signalo fazės charakteristiką kiekviename spinduliuojančiame sektorinės antenos elemente. Iš 5.13 pav. esančio grafiko galime teigti, jog susidaro didelis sukuriama EML energijos srauto tankio skirtumas, kai antena palenkama mechaniškai ar nusverinama elektriškai. Esant 4° palenkimui ties pagrindinio antenos kryptingumo charakteristikos lapelio sukuriama intensyvumo zona, esančia už 130 m nuo antenos, matome susidariusį 0,24 $\mu\text{W}/\text{cm}^2$ intensyvumą. Tokiu pačiu atstumu bei tokio pačio nuosvyrio, tik jau elektrinio sukuriamas 0,33 $\mu\text{W}/\text{cm}^2$ energijos srauto tankis. Kai antena mechaniškai palenkta 5°, tokiu pačiu atstumu susidaro 0,38 $\mu\text{W}/\text{cm}^2$ energijos srauto tankis, esant elektriniam palenkimui — 0,47 $\mu\text{W}/\text{cm}^2$ energijos srauto tankis.

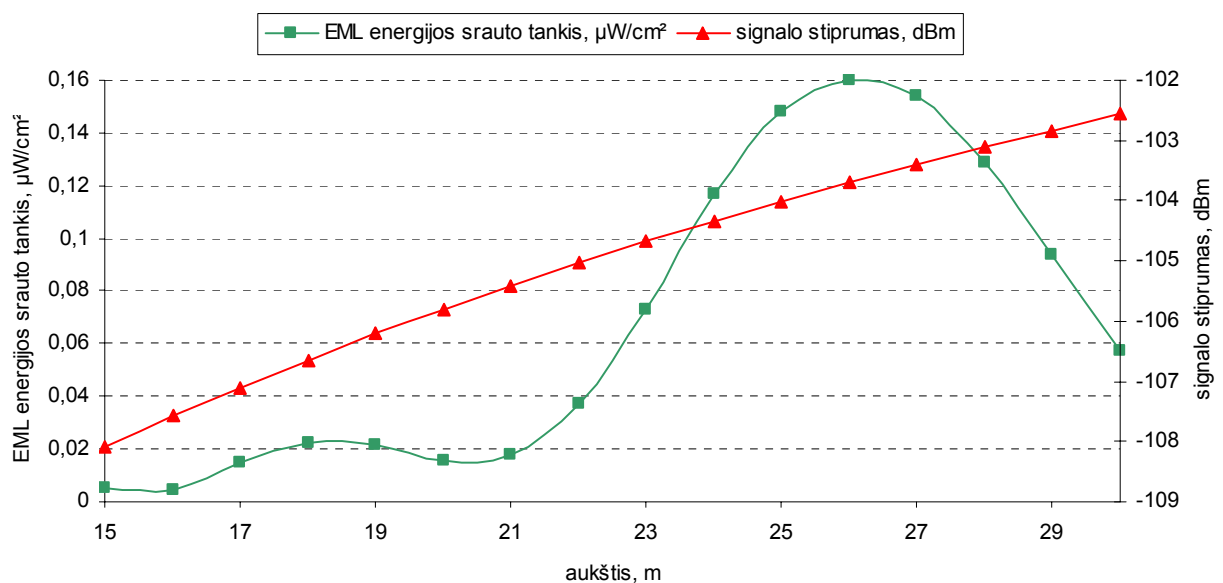
Reikia atkreipti dėmesį į šalutinius antenos kryptingumo charakteristikos lapelius, kurie susidaro mažu atstumu nuo antenos. Kuo didesnis mechaninis palenkimas, tuo šių lapelių sukuriama intensyvumo zonos traukiasi arčiau antenos. Elektrinio palenkimo atveju šie lapeliai išviso neegzistuoja ir 60 m atstumu nuo antenos sudaro 0,01 $\mu\text{W}/\text{cm}^2$ energijos srauto tankį, tuo tarpu mechaniškai 5° palenkta antena toje pat vietoje sukuria 0,96 $\mu\text{W}/\text{cm}^2$ energijos srauto tankį.

5.6. Energijos srauto tankio minimizavimo galimybės

Tyrime analizuojama, kaip besikeičiantys antenos ir siųstuvo parametrai lemia EML energijos srauto tankį artimoje radiotechninių objektų aplinkoje. Akivaizdu jog mažindami sukuriama intensyvumą greta antenos pakenksime paslaugos kokybei. Todėl šalia energijos srauto tankio kitimo nagrinėjama ir paslaugos kokybės kitimas, kuris vertinamas pagal signalo stiprumą.

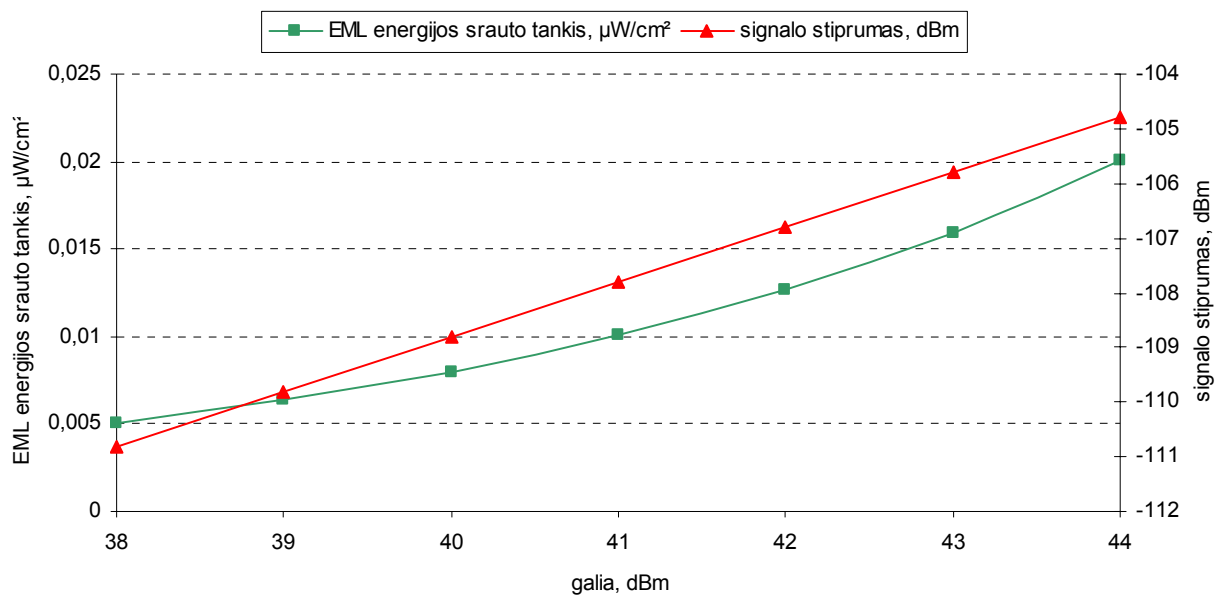
Signalų stiprumui tolimoje zonoje įvertinti naudojamas empirinis *Okumura-Hata* modelis, papildytas vidutinių dydžių miestų koeficientu. Skaičiavimai šiuo modeliu, keičiant tiriamus parametrus, atlikti 5 km atstumu nuo spinduliuojančio objekto, 1,5 m aukštyje.

Modeliuojant EML energijos srauto tankio sumažinimo galimybes buvo pasirinkti keturi keičiami parametrai — antenos aukštis, antenos palenkimas, antenos pasukimas bei siųstuvo galia. Keturi paminėti parametrai išnagrinėti atskirai, EML energijos srauto tankį skaičiuojant 1,5 m aukštyje, 50 m atstumu nuo antenos.



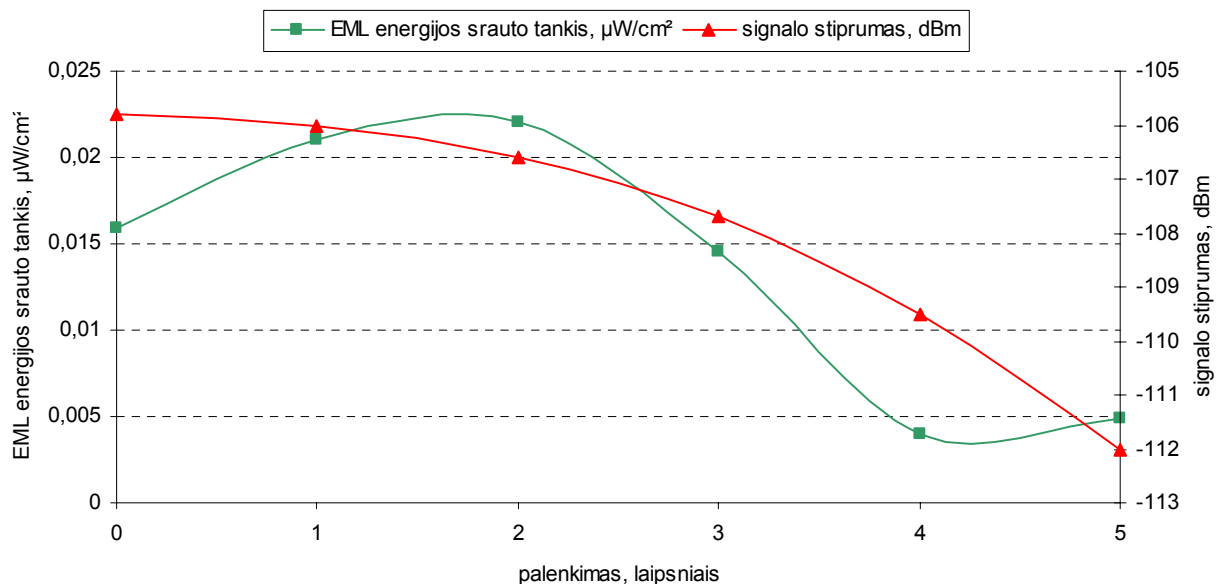
5.14 pav. Antenos aukščio įtaka EML energijos srauto tankiui bei signalo stiprumui

Iš grafiko matome, jog didėjant antenos aukščiui, tolimalame lauke tiesiškai didėja signalo stiprumas (5.14 pav.). Papildomai vienu metru pakelta antena signalo stiprumą pagerina $\sim 0,5$ dB — kai antenos aukštis 20 m, tai signalo stiprumas $-105,8$ dBm, kai antenos aukštis 30 m $-102,6$ dBm. Artimoje aplinkoje apskaičiuotas spinduliavimo intensyvumas kinta netiesiškai. Tai įtakoja antenos spinduliavimo diagrama. Matome, jog antenai esant 20 m aukštyje EML energijos srauto tankius siekia $0,016 \mu\text{W}/\text{cm}^2$, kai antena 3 metrai žemiau — $0,015 \mu\text{W}/\text{cm}^2$, kai 3 metrai aukščiau — $0,073 \mu\text{W}/\text{cm}^2$.



5.15 pav. Siųstuvo galios įtaka EML energijos srauto tankiui bei signalo stiprumui

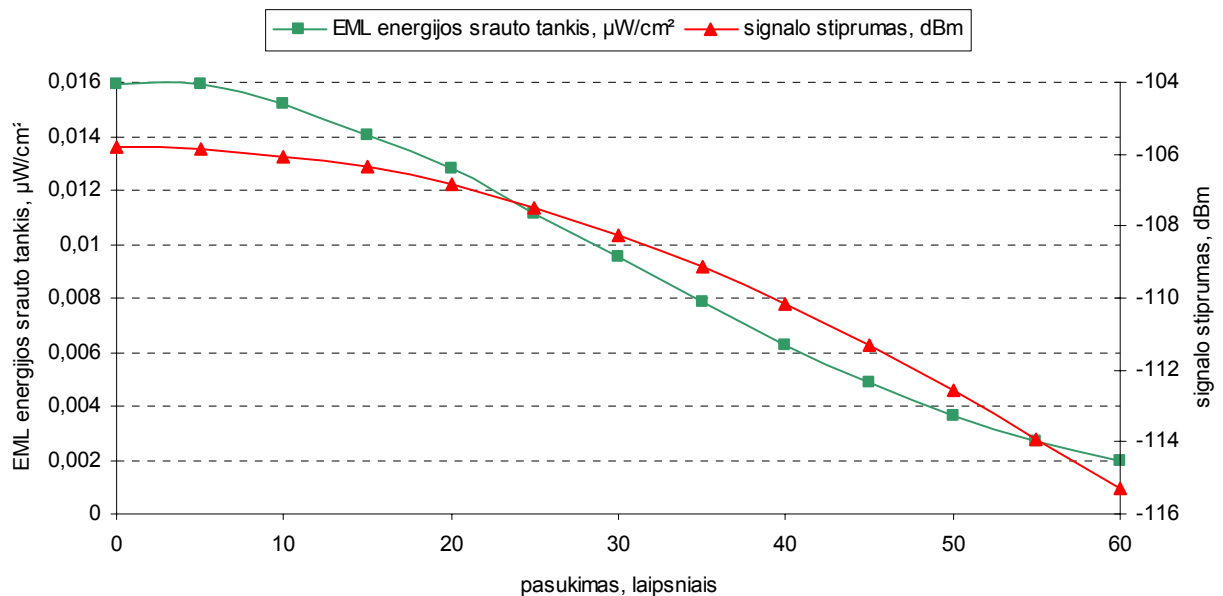
EML energijos srauto tankio bei signalo stiprumo priklausomybė nuo siųstuvo galios aiškiai išreikšta ir yra tiesinė (5.15 pav.). Siųstuvo galią padidinus 1 dB, signalo stiprumas irgi padidės vienu decibelu. Lyginant energijos srauto tankį ir siųstuvo galią gautas santykis — 1,25. Taigi, padidinus siųstuvo galią nuo 43 iki 44 dB intensyvumas padidės nuo 0,016 $\mu\text{W}/\text{cm}^2$ iki 0,02 $\mu\text{W}/\text{cm}^2$.



5.16 pav. Antenos palenkimo įtaka EML energijos srauto tankiui bei signalo stiprumui

Keičiant antenos mechaninį palenkimą matome, jog tolimajame lauke vyrauja tiesinis kitimas (5.16 pav.). Kadangi kampas tarp antenos ir skaičiuojamo taško už 5 km yra labai mažas, tai

kiekvienas papildomas laipsnis palenkiant anteną įtakoja signalo silpnėjimą tiek, kiek keičiasi antenos kryptingumo diagrama. Tokiu pačiu principu kinta ir energijos srauto tankis artimoje aplinkoje. Kitimas čia labiau išreikštas, nes kampas į anteną nuo matuojamo taško yra didesnis. Tokiu kampu ryškėja šalutinių antenos kryptingumo charakteristikos lapelių įtaka.

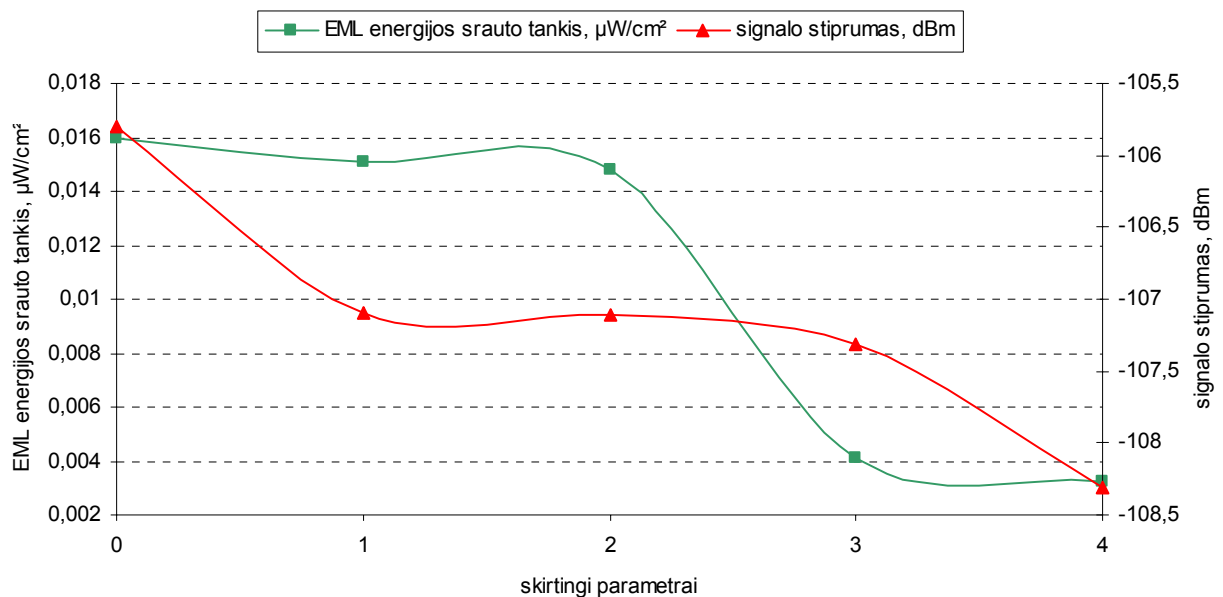


5.17 pav. Antenos pasukimo įtaka EML energijos srauto tankiui bei signalo stiprumui

Sukant anteną aplink vertikalią ašį spinduliavimo intensyvumas didžiausio spinduliavimo kryptimi nekinta. Atsižvelgiant į tai, jog sektorinė antena yra kryptinga, ją galima pakreipti taip, kad artimos zonos, kuriose pageidaujamas mažas energijos srauto tankis nepatektų į pagrindinio antenos kryptingumo charakteristikos lapelio spinduliavimo zoną (5.17 pav.). Iš grafiko matyti, jog pakreipus anteną 5° intensyvumas artimajame lauke išlieka pastovus $0,16 \mu\text{W}/\text{cm}^2$. Kreipiant anteną toliau intensyvumas tiesiškai mažėja — ties 20° siekia $0,013 \mu\text{W}/\text{cm}^2$, ties 30° $0,01 \mu\text{W}/\text{cm}^2$, ties 40° $0,006 \mu\text{W}/\text{cm}^2$. Tolimajame lauke signalo stiprumas irgi kinta tiesiškai, kuo didesnis antenos pakreipimas, tuo mažesnis signalo lygis susidaro.

Modeliuojamo radiotechninio objekto parametrus keičiant kompleksiskai galima sumažinti EML energijos srauto tankį artimoje aplinkoje. Sukurtos programos algoritmui nurodomos maksimalios ir minimalios parametrų vertės kurias reikės patikrinti su pasirinktu radiotechniniu objektu. Programa apskaičiavus galimas vertes pateikia EML energijos srauto tankio sumažinimo rezultatus. 5.18 pav. skaičiavimai atlikti $1,5 \text{ m}$ aukštyje, 50 m atstumu nuo antenos vertinant energijos srauto tankį ir 5 km atstumu nuo antenos vertinant signalo stiprumą. Iš grafiko matome,

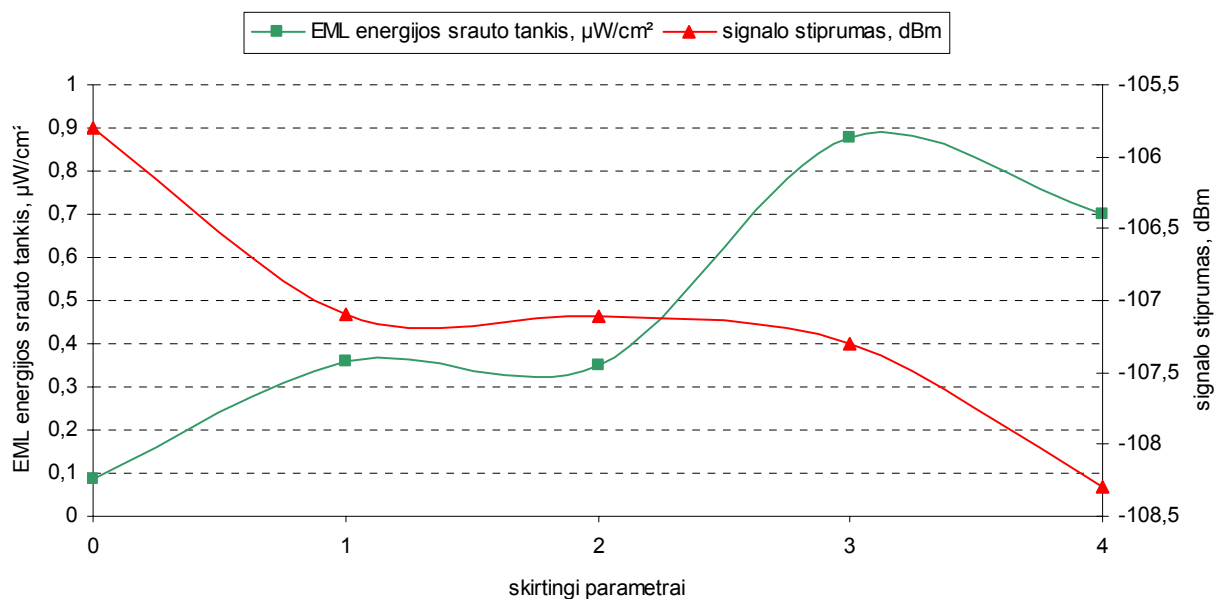
kaip keičiasi spinduliavimo intensyvumas artimoje aplinkoje bei signalo lygis tolimoje zonoje. Esant pradiniam siųstuvo bei antenos parametrams gaunamas $0,016 \mu\text{W}/\text{cm}^2$ energijos srauto tankis, bei $-105,8 \text{ dBm}$ signalo stiprumas, matomi ties „0“ abscisių ašyje. Atlikus modeliavimą buvo rasti parametrai, kurie mažiausiai nutolę nuo pradinės reikšmės tačiau galutiniame variante ženkliai lemia spinduliavimo intensyvumo sumažėjimą. Apskaičiuota, jog antenos perkėlimas 3 m žemiau stipriai pakeitė signalo stiprumą, kuris sumažėjo iki $-107,1 \text{ dBm}$ ir tik maža dalimi sumažino energijos srauto tankį iki $0,015 \mu\text{W}/\text{cm}^2$. Antrasis pakeistas parametras — antenos pakreipimas 5° abiejų rezultatų beveik neįtakoję. Tačiau pakeitus trečiąjį parametras, palenkus anteną 1° — energijos srauto tankį ženkliai sumažino iki $0,004 \mu\text{W}/\text{cm}^2$ ir tik nežymiai pakeitė signalo stiprumą tolimajame lauke iki $-107,3 \text{ dBm}$. Siųstuvo galios sumažinimas 1 dB iki 42 dBm labai nežymiai sumažino spinduliavimo intensyvumą artimoje zonoje iki $0,003 \mu\text{W}/\text{cm}^2$. Signalo stiprumas, tiesiškai priklausantis nuo siųstuvo galios irgi sumažėjo 1 dB. Iš grafiko matome, jog EML energijos srauto tankis pritaikius visus keturis parametrus sumažėjo 81 % nuo $0,016 \mu\text{W}/\text{cm}^2$ iki $0,003 \mu\text{W}/\text{cm}^2$, signalo stiprumas tolimoje zonoje pablogėjo 2,4 %, nuo $-105,8 \text{ dBm}$ iki $-108,3 \text{ dBm}$.



5.18 pav. EML energijos srauto tankio sumažinimas 1,5 m aukštyje, palaipsniui keičiant antenos
1) aukštį, 2) pasukimą, 3) palenkimą ir 4) siųstuvo galią

Atlikti modeliavimai 12 m aukštyje artimajame lauke, pritaikant geriausius rastus parametrus 1,5 m aukštyje. 5.19 pav. matome, kad norimas rezultatas netenkina, situacija pablogėjo — EML energijos srauto tankis padidėjo, signalo galia sumažėjo. 12 m aukštyje buvo apskaičiuota $0,09$

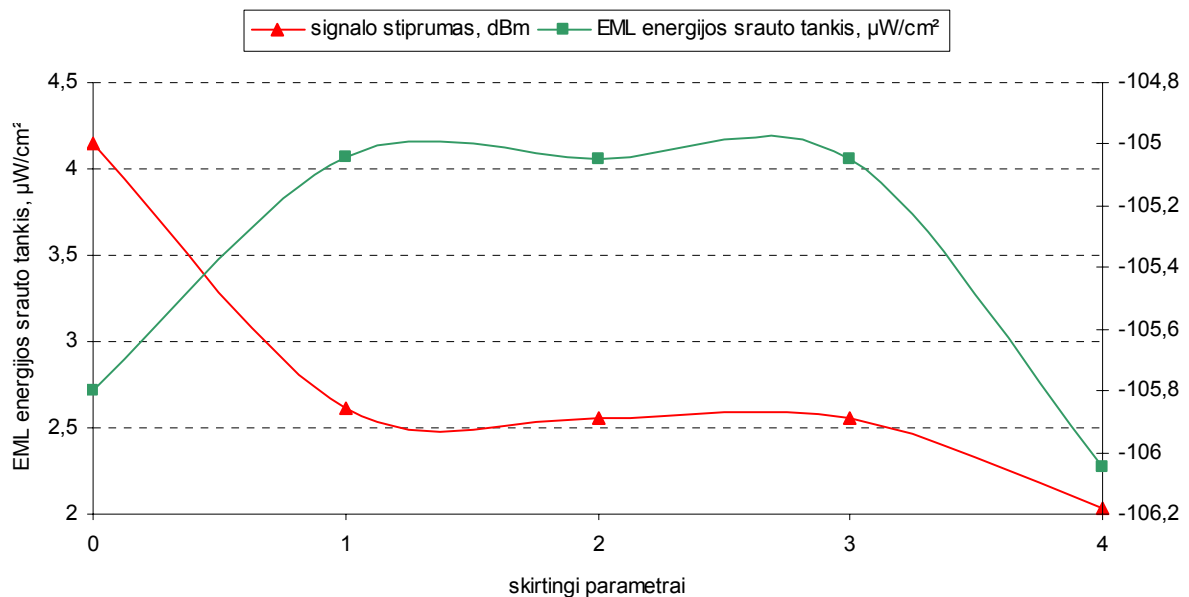
$\mu\text{W}/\text{cm}^2$ spinduliavimo intensyvumas. Pirmuoju atveju pakeitus antenos vietą 3 metrais žemiau gavome $0,36 \mu\text{W}/\text{cm}^2$, antenos pakreipimas stipriai situacijos nepakeitė — $0,35 \mu\text{W}/\text{cm}^2$. Antenos palenkimas padidino EML energijos srauto tankį iki $0,88 \mu\text{W}/\text{cm}^2$ ir tik tiesioginę priklausomybę su spinduliavimo intensyvumu turinti siųstuvo galia sumažino jį iki $0,7 \mu\text{W}/\text{cm}^2$. Pritaikius visus keturis parametrus, kurie ankstesniame modeliavime ženkliai sumažino spinduliavimo intensyvumą artimoje aplinkoje šiuo atveju jį tik padidino aštuonis kartus, nuo $0,09 \mu\text{W}/\text{cm}^2$ iki $0,7 \mu\text{W}/\text{cm}^2$. Signalų stiprumas tolimajame lauke sumažėjo nuo $-105,8 \text{ dBm}$ iki $-108,3 \text{ dBm}$.



5.19 pav. EML energijos srauto tankio kitimas 12 m aukštyje palaipsniui keičiant antenos
1) aukštį, 2) pasukimą, 3) palenkimą ir 4) siųstuvo galią

5.20 pav. matome kaip kinta EML energijos srauto tankis skaičiavimus atliekant 19 m aukštyje artimoje aplinkoje. Tolimasis laukas vertinamas kaip ir ankstesniuose modeliavimuose — 1,5 m aukštyje nutolus 5 km nuo antenos. Matome, jog antena 50 m atstumu, 19 m aukštyje sukuria $4,2 \mu\text{W}/\text{cm}^2$ energijos srauto tankį. Programa ieškanti geriausių parametų, antenos aukštį padidino 2 m iki 22 m. Spinduliavimo intensyvumas stipriai sumažėjo ir siekia $2,6 \mu\text{W}/\text{cm}^2$. Signalų stiprumas padidinus antenos aukštį tolimoje zonoje pagerėjo nuo $-105,8 \text{ dBm}$ iki -105 dBm . Pakreipus anteną aplink savo vertikalią ašį 5° ženkliai nepakito nei spinduliavimo intensyvumas ($2,56 \mu\text{W}/\text{cm}^2$), nei signalų stiprumas. Tikrinimą atliekanti programa nerado mažesnių energijos srauto tankio verčių keičiant antenos palenkimą, todėl antenos palenkimas liko 0° ir tai niekaip neįtakoją skaičiuojamų verčių. EML energijos srauto tankis ir signalų stiprumas tiesiškai priklauso nuo siųstuvo galios. Ties abscisių ašyje esančiu „4“ parametru matome, kad siųstuvo galią

sumažinus 1 dB, spinduliavimo intensyvumas sumažėjo iki $2,04 \mu\text{W}/\text{cm}^2$, o signalo stiprumas nuo -105 dBm iki -106 dBm .



5.20 pav. EML energijos srauto tankio sumažėjimas 19 m aukštyje palaipsniui keičiant antenos
1) aukštį, 2) pasukimą, 3) palenkimą ir 4) siųstuvo galią

EML energijos srauto tankis 5.20 pav. pritaikius visus keturis parametrus sumažėjo 51 % nuo $4,2 \mu\text{W}/\text{cm}^2$ iki $2,04 \mu\text{W}/\text{cm}^2$. Signalų stiprumas tolimajame lauke pasikeitė nuo $-105,8 \text{ dBm}$ iki -106 dBm . Tokį mažą šio parametro pokytį įtakoją tai, kad antenos aukštis buvo padidintas ir dėlto tolimajame lauke nesusidarė tokie dideli signalo nuostoliai.

5.6. Išvados

Skyriuje aptarta skirtingų antenos bei siųstuvo parametrų įtaka EML energijos srauto tankiui. Pagrindiniai parametrai — antenos aukštis, antenos palenkimas, antenos kryptis bei siųstuvo galia sumodeliuoti atskirai. Ištirta šių parametrų įtaka artimoje spinduliavimo zonoje.

Ištirta, kad EML energijos srauto tankis labai priklauso nuo antenos aukščio. Aptartos antenos modeliavimo metu buvo nustatyta, jog keičiant antenos aukštį keičiasi ne tik energijos srauto tankio dydis, bet ir atstumas už kurio susidaro maksimalios vertės. Buvo pastebėta, kad keičiant antenos aukštį, jos sukuriama tokio pačio EML energijos srauto tankio plotas laipsniškai kinta — kuo antena arčiau skaičiuojamo taško, tuo tokio paties intensyvumo dengiamas plotas didesnis. Atlikus skaičiavimus keliuose aukščiuose skirtingu atstumu nuo antenos matyti, kad kuo didesnis atstumas, tuo antenos kryptingumo charakteristika mažiau lemia susidarantį energijos srauto tankį.

Modeliuojant antenos mechaninį palenkimą nustatyta, kad didėjant palenkimo kampui mažėja atstumas, už kurio pagrindinis antenos kryptingumo charakteristikos lapelis sukuria intensyvią zoną. Kryptingumo diagramoje esančių minimumų dėka palenkimą sumažinus dviem laipsniais intensyviausioje spinduliavimo zonoje galime gauti net per pus mažesnę energijos srauto tankį. Sulyginus tokio paties kampo mechaninio ir elektrinio nuosvyrio sukuriamą spinduliavimo intensyvumą nustatyta, kad esant elektriniam palenkimui sukuriamas daug didesnis energijos srauto tankis tame pačiame taške.

Pagal atliktus skaičiavimus keičiant siųstuvo galią matyti, jog EML energijos srauto tankio priklausomybė nuo siųstuvo galios yra tiesinė tiek šalutinio antenos kryptingumo charakteristikos lapelio zonoje, tiek ir pagrindinio antenos kryptingumo charakteristikos lapelio zonoje. Apskaičiavus spinduliavimo intensyvumo pasiskirstymą didesniu atstumu nuo antenos matome, kad radiotechniniai objektai esantys toliau nei 300 m atstumu nuo tiriamojo taško sukuria labai mažo lygio EML energijos srauto tankį.

Papildomai vienu metru pakelta antena signalo stiprumą padidina $\sim 0,5$ dB, matuojant 5 km atstumu. Artimoje aplinkoje apskaičiuotas spinduliavimo intensyvumas kinta netiesiškai. Tai įtakoja antenos spinduliavimo diagrama. EML energijos srauto tankio bei signalo stiprumo priklausomybė nuo siųstuvo galios aiškiai išreikšta ir yra tiesinė. Lyginant energijos srauto tankį ir siųstuvo galią buvo gautas santykis — 1,25. Keičiant antenos palenkimą matome, jog tolimajame lauke vyrauja tiesinis kitimas. Artimoje aplinkoje, kitimas labiau išreikštas, nes pradeda ryškėti šalutinių antenos kryptingumo charakteristikos lapelių įtaka. Dėl sektorinės antenos kryptingumo savybių, ją galima pakreipti taip, kad artimos zonos, kuriose pageidaujamas mažas energijos srauto tankis nepatektų į pagrindinio antenos kryptingumo charakteristikos lapelio spinduliavimo zoną.

Buvo nustatyta, jog modeliujamo radiotechninio objekto parametrus keičiant kompleksiskai EML energijos srauto tankį artimoje aplinkoje galima reguliuoti daug lanksčiau. Skaičiavimai parodė, kad EML energijos srauto tankis taikant visus keturis parametrus gali sumažėti net 81 % pasirinktame taške. Signalio stiprumas tolimoje zonoje visais modeliavimo atvejais sumažėjo nežymiai, daugiausia 2,4 %. Iš pateiktų grafikų matome, jog negalime vadovautis vienodais antenos ir siųstuvo parametrais norint sumažinti spinduliavimo intensyvumą skirtinguose taškuose. Pritaikius visus keturis parametrus, kurie ankstesniame modeliavime ženkliai sumažino spinduliavimo intensyvumą artimoje aplinkoje kitame EML energijos srauto tankį padidino aštuonis kartus.

Atlikti modeliavimai parodė, kad norint sumažinti EML energijos srauto tankį artimoje aplinkoje neatliekant papildomų skaičiavimų galima tik mažinant siųstuvo galią. Keičiant antenos aukštį, palenkimą ar kryptį spinduliavimo intensyvumui didelę įtaką daro antenos kryptingumo

charakteristika. Todėl pakeitus šiuos parametrus kas kart būtina iš naujo įvertinti visus svarbius taškus ir bendrą sukuriamą draudžiamąją zoną. Norint sumažinti energijos srauto tankį greta antenos reikia atsižvelgti į tai, kokia antenos kryptingumo charakteristika. Galima parinkti anteną, kurioje šalutiniai antenos kryptingumo charakteristikos lapeliai yra minimizuoti.

6. IŠVADOS

Darbe išnagrinėtas Lietuvoje naudojamas modelis elektromagnetinių laukų intensyvumui vertinti ir empirinis *Okumura-Hata* modelis. Atliktas eksperimentinis realios antenos sukuriama elektromagnetinio lauko intensyvumo matavimas. Sukurta taikomoji programa, kuri įvertina spinduliavimo intensyvumą artimoje aplinkoje. Naudojantis programa atlikti modeliavimai, iš kurių matyti, kokiomis priemonėmis galima sumažinti EML energijos srauto tankį artimoje aplinkoje.

Pasinaudojus modelio EML intensyvumo parametrų įvertinimo metodika buvo atlikti teoriniai esamos antenos sukuriama EML intensyvumo skaičiavimai. Teoriniai skaičiavimai parodė, jog esant 15 m atstumu ir 3 m žemiau už siųstuvo horizontaliai sumontuotą antena, EML intensyvumas neviršija higienos normose leidžiamų lygių, o artimesnėje kaip 15 m zonoje intensyvumas keliose vietose viršijamas iki dviejų kartų. Eksperimentinis matavimas buvo atliktas su atestuotu matavimo prietaisu EMR–300. Rezultatai parodė, kad EML energijos srauto tankis leidžiamą lygį viršija nutolus nuo 30 m iki 50 m atstumu.

Modeliavimo uždaviniams palengvinti buvo sukurta programa, kuri automatizuoja skaičiavimus. Naudojantis sukurta programa galima sumodeliuoti neribojamo kiekio radiotechninių objektų sukuriamus suminius laukus, bei įvertinti sukuriamas draudžiamąsias zonas. Gauti rezultatai pateikiami programos lange esančiame žemėlapyje, vertes galima eksportuoti polinėje koordinatų sistemoje. Programoje įdiegta galimybė, automatiškai keisti parametrus siekiant sumažinti spinduliavimo intensyvumą pasirinktuose taškuose. Radiotechninių objektų duomenis galima keisti grafinėje vartotojo aplinkoje, taip pat kaip ir nustatyti objekto koordinates žemėlapyje. Programa gali būti panaudojama mokymo procese.

Ištirta, kad keičiant antenos aukštį, jos sukuriamas tokio pačio EML energijos srauto tankio plotas laipsniškai kinta — kuo antena arčiau skaičiuojamo taško, tuo tokio paties intensyvumo dengiamas plotas didesnis. Artimoje aplinkoje keičiant antenos aukštį apskaičiuotas spinduliavimo intensyvumas kinta netiesiškai. Tai nulemia antenos spinduliavimo diagrama. Nustatyta, kad EML energijos srauto tankio bei signalo stiprumo priklausomybė nuo siųstuvo galios aiškiai išreikšta ir yra tiesinė. Lyginant energijos srauto tankį ir siųstuvo galią buvo gautas santykis — 1,25. Keičiant antenos palenkimą matome, jog tolimajame lauke vyrauja tiesinis kitimas. Sektorines antenas, pasižyminčias kryptingumu galima pakreipti taip, kad artimos zonos, kuriose pageidaujamas mažas energijos srauto tankis nepatektų į pagrindinio antenos kryptingumo charakteristikos lapelio spinduliavimo zoną.

Buvo nustatyta, jog modeliuojamo radiotechninio objekto parametrus keičiant kompleksiskai EML energijos srauto tankis artimoje aplinkoje gali būti reguliuojamas daug lanksčiau. Skaičiavimai parodė, kad EML energijos srauto tankis taikant visus keturis parametrus gali sumažėti net 81 % pasirinktame taške. Signalų stiprumas tolimoje zonoje visais modeliavimo atvejais sumažėjo nežymiai, daugiausia 2,4 %. Iš pateiktų grafikų matome, jog negalime vadovautis vienodais antenos ir siųstuvo parametrais norint sumažinti spinduliavimo intensyvumą skirtinguose taškuose. Pritaikius visus keturis parametrus, kurie ankstesniame modeliavime ženkliai sumažino spinduliavimo intensyvumą artimoje aplinkoje, kitame EML energijos srauto tankį padidino aštuonis kartus.

Atlikti modeliavimai parodė, kad norint sumažinti EML energijos srauto tankį artimoje aplinkoje, neatliekant papildomų skaičiavimų, galima tik mažinant siųstuvo galią. Keičiant antenos aukštį, palenkimą ar kryptį spinduliavimo intensyvumui didelę įtaką daro antenos kryptingumo charakteristika. Todėl pakeitus šiuos parametrus kas kart būtina iš naujo įvertinti visus svarbius taškus ir bendrą sukuriamą draudžiamąją zoną. Norint sumažinti energijos srauto tankį greta antenos reikia atsižvelgti į tai, kokia antenos kryptingumo charakteristika. Galima parinkti anteną, kurioje šalutiniai antenos kryptingumo charakteristikos lapeliai yra minimizuoti.

Šis darbas ateityje galėtų būti pratęstas papildant atliktus tyrimus. Tikslinga pagerinti mažesnio spinduliavimo intensyvumo apskaičiavimo programą įvedant ne tik svarbius taškus, bet ir svarbius plotus.

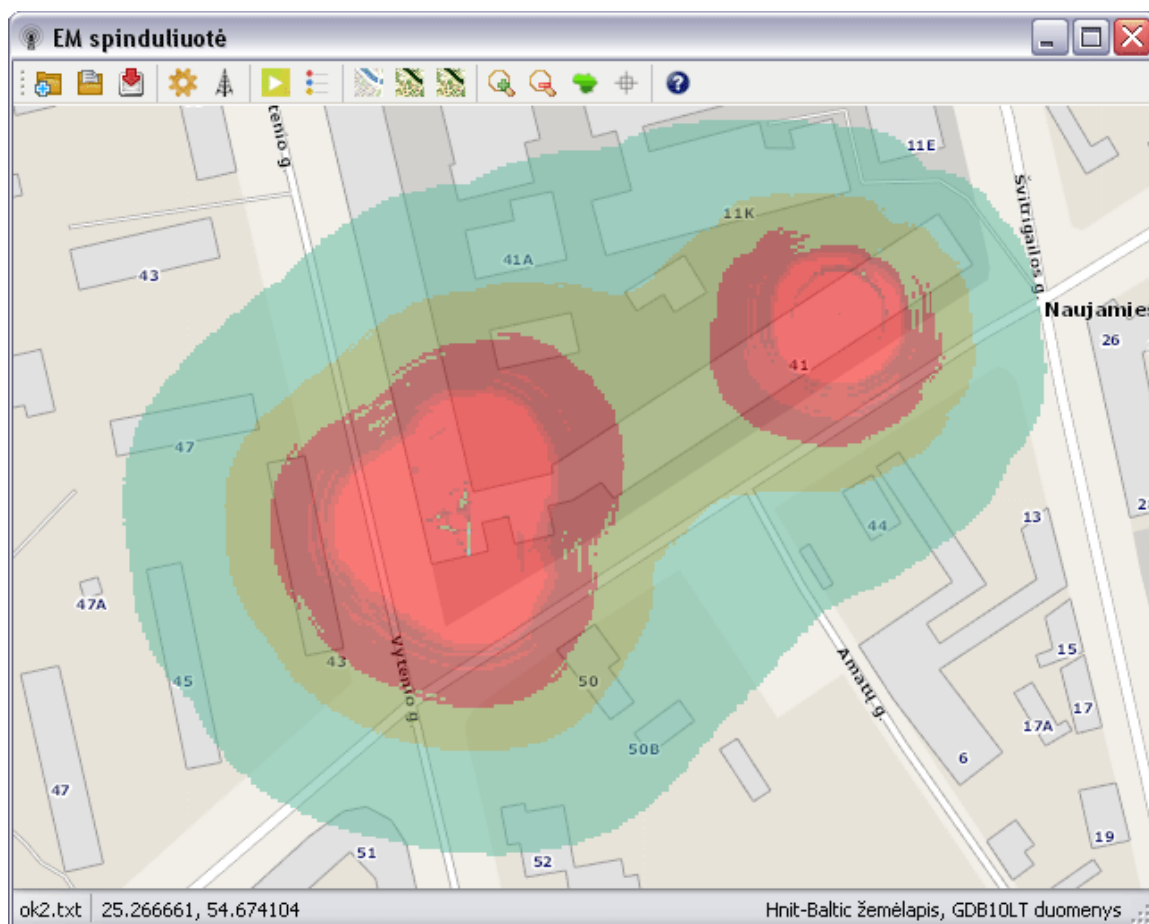
7. LITERATŪROS SĄRAŠAS

1. HN 80:2011 *Elektromagnetinis laukas darbo vietose ir gyvenamojoje aplinkoje. Parametrų normuojamos vertės ir matavimo reikalavimai 10 kHz – 300 GHz radijo dažnių juostoje* (Žin., 2011, Nr. 29-1374, Nr. 39-1896).
2. *Dėl radiotechninio objekto radiotechninės dalies projekto ir elektromagnetinės spinduliuotės stebėsenos plano derinimo tvarkos aprašo patvirtinimo*. Lietuvos Respublikos sveikatos apsaugos ministro įsakymas, 2011 m. kovo 2 d. Nr. V-200.
3. Pocius R.V. *Radiotechninių objektų elektromagnetinės ekspertizės klausimai*. Elektronika ir elektrotechnika, 1(36). 2002, p. 38–42.
4. *Nacionalinė radijo dažnių paskirstymo lentelė*. RRT. 2011.
5. Mishra, A.R. *Fundamentals of Cellular Network Planning and Optimization*. England: John Wiley & Sons. 2004, 307 p.
6. Holma H., Toskala A. *WCDMA for UMTS. Radio Access for Third Generation Mobile Communications. Third Edition*. England: John Wiley & Sons. 2004, 450 p.
7. Dutta A., Sarin S. *Performance optimization in GSM networks through dynamic power control*. High Speed Networks and Multimedia Communications 5th IEEE International Conference on. 2002, p. 197–201.
8. Rohner C. *Antenna basics*. Rohde&Schwarz. 2006, 76 p.
9. Scholz P. *Basic antenna principles for mobile communications*. Kathrein-Werke KG. 46 p.
10. TN1200.04. *Calculating radiated power and field strength for conducted power measurements*. Semtech. 2007, 9 p.
11. *ICNIRP guidelines for limiting exposure to time varying electric, magnetic and electromagnetic fields (up to 300 GHz)*. Health physics, 74(4). 1998, p. 494–522.
12. Tamošiūnienė E., Urbonas M., Mačiūnas E. *Nejonizuojančios spinduliuotės valdymo srities teisinės bazės, įskaitant teritorijų planavimą ir statybų procesą, analizė*. Veiklos ataskaita Nr. Pa-2. 2010, 25 p.
13. Europos Tarybos 1999 m. liepos 12 d. *Rekomendacija 1999/519/EB dėl elektromagnetinių laukų (nuo 0 iki 300 GHz) poveikio žmonėms*. 1999, 59 p.
14. Anskaitis A., Guršnys D., Mikėnas K., Pocius R.V. *The analysis of the influence of adjacent radio stations positions on total electromagnetic field intensity observed near the projected BS*. Proceedings of the XXII international conference on Electromagnetic Disturbances. 2012, p. 6–9.

15. *Определение уровней электромагнитного поля, создаваемого излучающими техническими средствами телевидения, ЧМ радиовещания и базовых станций сухопутной подвижной радиосвязи. Методические указания МУК 4.3.1677-03.* 2003, 22 p.
16. Pocius R.V. *Antenų kryptingumo diagramų nuosvyrio įtaka formuojant mobiliojo korinio ryšio ląstelių ribas ir vertinant elektromagnetinio lauko intensyvumą.* Elektronika ir elektrotechnika, 8(64). 2005, p. 31–36.
17. Okumura Y., Ohmori E., Kagano T., Fukuda K. *Field Strength and Its Variability in VHF and UHF Land-Mobile Radio Service.* Review of the Electrical Communication Laboratory, 16, 1968, p. 825–873.
18. Hata, M. *Empirical Formulae for Propagation Loss in Land Mobile Radio Services.* IEEE Transactions Vehicular Technology, 29(3). 1980, p. 317–325.
19. Damasso E., Correia L.M. *COST 231. Digital Mobile Radio Towards Future Generation Systems.* Final Report, COST Telecom Secretariat. 1999, p. 134–148.
20. *EMR-20/-30, EMR-200/300 radiation meters for isotropic measurement of electromagnetic fields.* Operating manual. Eningen, Germany: Wandel&Goltermann; 1998, 140 p.
21. *Lietuvos skaitmeninis ortofotografinis M 1:10000 žemėlapis ORT10LT.* Nacionalinė žemės tarnyba prie ŽŪM, 2010.

A PRIEDAS. PROGRAMOS NAUDOJIMO APRAŠYMAS

Programa „EM spinduliuotė“, tai automatizuotas elektromagnetinio spinduliavimo intensyvumo įvertinimo įrankis, kuris gautus skaičiavimo rezultatus pateikia žemėlapyje (1 pav.).



1 pav. Programos fragmentas

A.1. Programos įrankių juosta

- | | |
|--|---|
|  — Sukurti naują projektą |  — Rodyti vektorinį žemėlapi |
|  — Atidaryti projektą |  — Rodyti orto nuotrauką |
|  — Išsaugoti projektą |  — Pritraukti žemėlapi |
|  — Programos nustatymai |  — Atitolinti žemėlapi |
|  — Radiotechninių objektų sąrašas |  — Rodyti visą žemėlapi |
|  — Pradėti skaičiavimą |  — Centruoti žemėlapi |
|  — Skaitiniai rezultatai |  — Programos informacija |

A.2. Žemėlapių navigacija

Kairiuoju pelės mygtuku nuspaudus ant žemėlapių, rodomą vaizdą galima perstumti. Tokiu būdu keisite programoje rodomo žemėlapių centro koordinatės. Taipogiai žingsniuoti galima naudojantis klaviatūros rodyklių <←>, <↑>, <→>, <↓> klavišais.

Pelės žymekliui esant programos žemėlapių ribose, pasinaudojus pelės ratuku galima pritraukti arba atitolinti žemėlapi. Tokį patį veiksmą galima atlikti ir įrankių juostoje esančių mygtukų   pagalba.

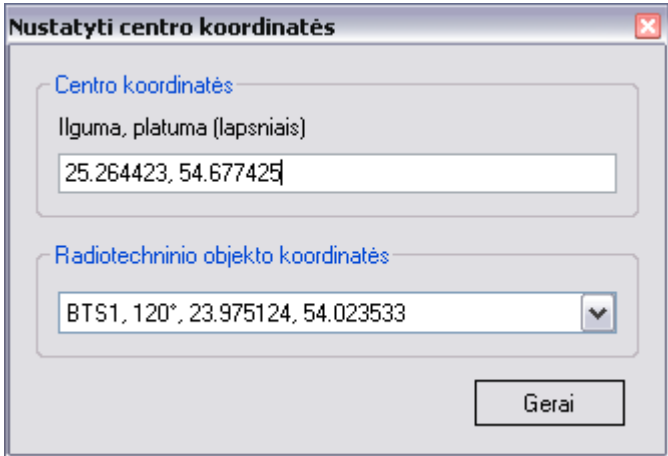
Norint vienu kartu stipriai atitolinti žemėlapi galima pasinaudoti įrankių juostos mygtuku .

Paspaudus žemėlapių centravimo mygtuką , bus parodytas kitas programos langas (2 pav.). Pastarajam langui atsidarius bus rodomos esamos žemėlapių centro koordinatės. Ten pat galima ranka įvesti naujas žemėlapių centro koordinatės, arba centruoti žemėlapi pagal pasirinktos bazinės stoties koordinatės.

Žemėlapių centro koordinatės ir pritraukimo lygis išsaugomas uždarant programą, todėl paleidus programą iš naujo, žemėlapyje bus rodoma ankstesnė vieta.

A.3. Žemėlapių sluoksniai

Šiuo metu programoje įdiegti trys žemėlapių sluoksniai. Įrankių juostos mygtuko  pagalba galima iškviešti vektorinio žemėlapių rodymą. Sekančių dviejų įrankių juostos mygtukų  pagalba galima įjungti orto nuotraukos rodymą. Pirmasis mygtukas įjungus 2010 m. leidybos orto nuotrauką, antrasis — 2005 m. leidybos. Taip pat tarp žemėlapių sluoksnių galima persijungti klaviatūros mygtukų <F5>, <F6>, <F7> pagalba.



2 pav. Žemėlapių centro koordinatės įvedimas.

Žemėlapiu sluoksnio pasirinkimas išsaugomas uždarant programą. Jeigu programa vėliau vėl bus paleista, bus rodomas anksčiau pasirinktas sluoksnis.

A.4. Programos nustatymai

Programoje galima naudotis dviejų koordinatinių sistemomis — Lietuvos koordinatinių sistema (LKS-94) bei Pasauline geodezinių koordinatinių sistema (WGS-84).

Atliekant modeliavimą analitiniu elektromagnetinių laukų intensyvumo įvertinimo artimoje bazinių stočių aplinkoje metodu, reikia nurodyti žemės paviršiaus poveikio koeficientą K_R , bei koeficientą, kuris įvertina nukrypimą nuo pagrindinės antenos spinduliavimo krypties horizontalioje plokštumoje K_H .

Intensyvumo modeliavimo rezultatus atvaizduojant žemėlapyje, reikia pasirinkti, kokiomis spalvomis bus žymimos apskaičiuoto intensyvumo zonos (4 pav.). Jeigu laukiami modeliavimo rezultatai nėra žinomi, minimalias $\mu\text{W}/\text{cm}^2$ vertes reikėtų nurodyti plačiame diapazone. Programa nereikalauja, tačiau tam kad pamatyti žemėlapyje atvaizduotus rezultatus, reikia nurodyti bent vieną minimalią vertę, kuri būtų mažesnė negu didžiausia sumodeliuota. Spalvą galima pakeisti paspaudus ant norimos spalvos kvadrato. Jeigu spalva nėra reikalinga — įveskite $0 \mu\text{W}/\text{cm}^2$.

Projekto nustatymai

Koordinatinių formatas

☐ LKS-94

☒ WGS-84 (laipsniais)

Modelio koeficientai

Koeficientas, įvertinantis žemės paviršiaus poveikį (K_R)

Koeficientas, įvertinantis nukrypimą nuo pagrindinės antenos spinduliavimo krypties horizontalioje plokštumoje (K_H)

Programos parametrai

☐ Atnaujinti stumiant

Intensyvumo spalvos

Spalva	Minimali vertė
	2 $\mu\text{W}/\text{cm}^2$
	4 $\mu\text{W}/\text{cm}^2$
	6 $\mu\text{W}/\text{cm}^2$
	8 $\mu\text{W}/\text{cm}^2$
	10 $\mu\text{W}/\text{cm}^2$

Spalvų užpildymas

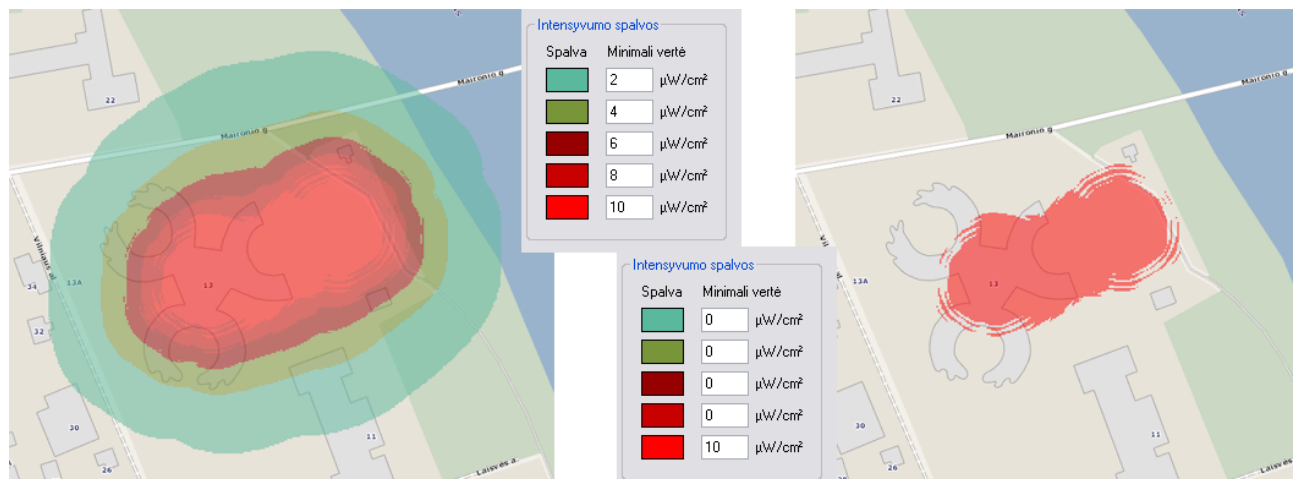
% (0% - permatoma)

Gera

3 pav. Programos nustatymų langas.

Rezultatų atvaizdavimo žemėlapyje intensyvumą galima nustatyti „spalvų užpildymo“ laukelyje. Įvedus 100 % rezultatų sluoksnis bus nepermatomas, įvedus 0 % rezultatų sluoksnio nesimatys. Puikus sprendimas — naudoti 50 % užpildymą.

Koordinatinių sistemos pakeitimai pradeda galioti iškart visoje programoje. Tuo tarpu norint pamatyti skirtumus įvedant intensyvumo zonų spalvas ar keičiant modeliavimo koeficientus — skaičiavimą reikia paleisti iš naujo. Visi šie nustatymai išjungiant programą yra išsaugojami ir kitą kartą programą paleidus rodomi nepakeisti.



4 pav. Atvaizduojami rezultatai, kai nustatytos penkios spalvinės zonos (kairėje), ir viena zona (dešinėje).

A.5. Radiotechniniai objektai

Radiotechninių objektų įvedimo lango kairėje yra visų įvestų radiotechninių objektų sąrašas. Sąraše esančių įrašų pavadinimai sudaromi iš stoties pavadinimo ir antenos krypties (5 pav.).

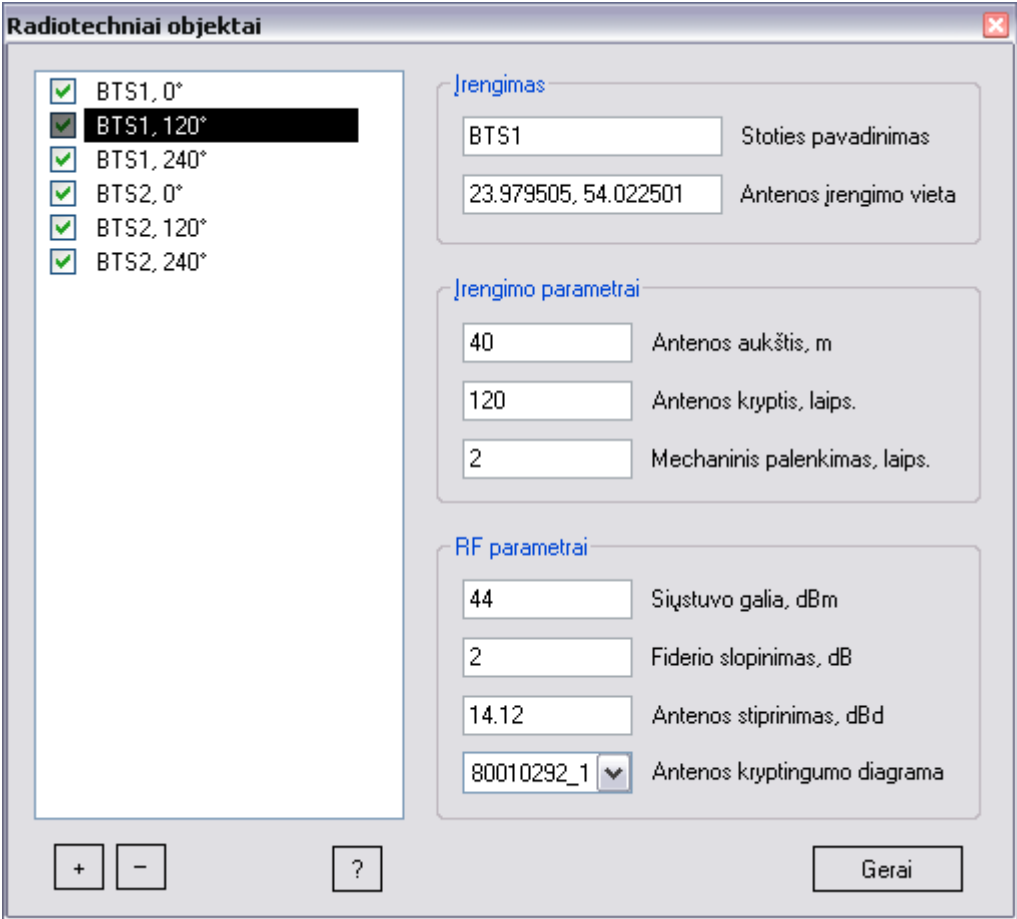
Tam, kad įvesti naują radiotechninį objektą, reikia pažymėti vieną iš jau esamų ir paspausti apačioje esantį mygtuką „+“. Taip atkartotas ankstesnis radiotechninis objektas. Įrašo pašalinimui reikia pasirinkti norimą radiotechninį objektą ir paspausti mygtuką „-“.

Kiekvienas radiotechninis objektas turi savo parametrų vertes. Paspaudus sąraše ant norimo radiotechninio objekto, dešinėje esanti lentelė užpildoma pastarojo objekto informacija, kurią galima keisti.

Radiotechninio objekto koordinatės galima įvesti ranka arba pažymėti koordinatės tiesiai žemėlapyje. Pasirinkite norimą keisti radiotechninį objektą ir paspauskite mygtuką „?“ — redagavimo langas užsidarys, o žemėlapis bus išcentruotas ties norimu keisti objektu. Paspausdami dešinįjį pelės mygtuką pasirinkite norimą naują objekto vietą. Kai nauja vieta tenkins —

paspauskite įrankių juostos mygtuką . Sugrįšite į redagavimo langą, kuriame jau bus atnaujintos objekto koordinatės.

Antenų kryptingumo diagramų failai saugomi programos direktorijoje esančiame aplanke „antenos“. Pastarąjį aplanką reikėtų papildyti norimų naudoti antenų kryptingumo diagramų failais, kurių formatas „Planet / MSI“.



5 pav. Radiotechninių objektų sąrašas ir jų nustatymai.

Norint, kad kuris nors radiotechninis objektas nepakliūtų į bendrą modeliavimą, tereikia nuimti varnelę esančią ties jo pavadinimu sąrašė.

A.6. Duomenų įkrovimas ir išsaugojimas

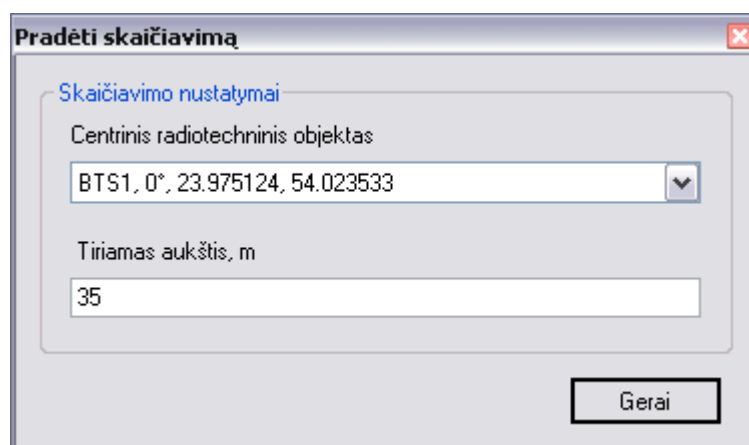
Norint sukurti naują modeliavimo projektą, programos įrankių juostoje reikia nuspausti  mygtuką. Jeigu norite atidaryti ankstesnį projektą su radiotechninių objektų duomenimis, paspauskite  mygtuką programos įrankių juostoje — galėsite pasirinkti norimą duomenų failą.

Įvestus naujai, ar pakeistus duomenis pravartu išsaugoti, tai galima padaryti  mygtuko pagalba. Radiotechninių objektų duomenis patartina išsaugoti po kiekvieno pakeitimo prieš atliekant skaičiavimo užduotį. Jeigu mygtuką  paspausite tuo metu, kai laikysite nuspaudę klaviatūros klavišą <shift>, galėsite radiotechninių objektų duomenų failą išsaugoti kitu vardu.

A.7. Rezultatų modeliavimas

Kai visos norimos modeliuoti bazinės stotys įvestos pasinaudojant radiotechninių objektų įvedimo langu, galima atlikti intensyvumo skaičiavimo užduotį.

Programos įrankių juostoje paspaudus  mygtuką bus parodytas langas (6 pav.), kuriame reikės nurodyti modeliavime tiriamą aukštį, bei kurio radiotechninio objekto atžvilgiu bus atliekamas pastarasis modeliavimas.



6 pav. Rezultatų skaičiavimo pradžios langas

Pastarajam langui užsidarius modeliavimas baigiamas ir žemėlapyje pateikiamos intensyvumo zonos (4 pav.), pagal nustatymuose pasirinktus parametrus.

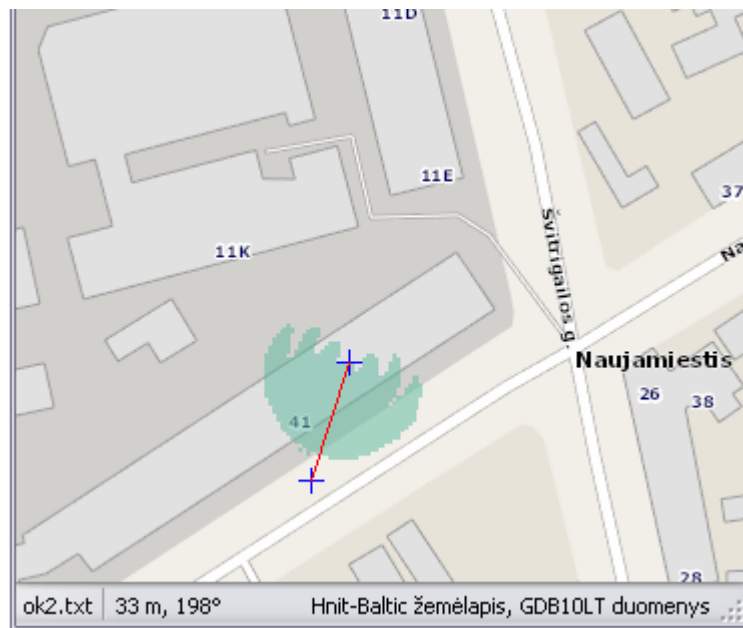
A.8. Gauti rezultatai

Atlikus modeliavimą, intensyvumo zonas galima analizuoti ne tik grafiškai, žemėlapyje, bet ir atlikti skaitinę analizę. Programos įrankių juostoje nuspaudus  mygtuką bus atvertas langas, kuriame matomi modeliavimo rezultatai. Rezultatai pateikiami polinėj koordinačių sistemoje, kas 5°, iki 300 m atstumu. Apskaičiuotas intensyvumas, kuris viršija šiuo metu galiojančią higienos normą ($10 \mu\text{W}/\text{cm}^2$), pažymimas raudonai. Gautus rezultatus polinėje sistemoje tolimesniems

tyrimams, galima eksportuoti į „Microsoft Excel“ dokumentą — reikia nuspausti lango apačioje esantį mygtuką „Eksportuoti“.

A.9. Papildomi įrankiai

Programa turi atstumo matavimo funkciją (7 pav.). Norimoje vietoje paspaudus dešinę pelės mygtuką, žemėlapyje atsiras žymeklis, kuris reikš matavimo pradžios tašką. Jeigu nuspausite ir laikysite klaviatūros klavišą <shift>, kol dar kartą spausite dešinę pelės mygtuką — bus išmatuotas atstumas. Programos apačioje esančioje būsenos juostoje rodomas išmatuotas atstumas bei kryptis.



7 pav. Atstumo matavimo funkcija

A.10. Išvados

Naudojantis sukurta programa galima sumodeliuoti neribojamą kiekį radiotechninių objektų. Gauti rezultatai bus pateikiami programos lange esančiame žemėlapyje Dekarto koordinatų sistemoje, bei duomenų lentelėje, polinėje koordinatų sistemoje. Radiotechninių objektų duomenis galima keisti grafinėje vartotojo aplinkoje, taip pat kaip ir nustatyti objekto koordinates žemėlapyje. Elektromagnetinio lauko intensyvumo įvertinimo programa papildyta geografinių koordinatų keitimo įrankiu bei atstumų ir krypties matuokle.

Elektromagnetinio lauko intensyvumo

MATAVIMO AKTAS

2012–08–26

Siųstuvo dažnis: 432 MHz

Siųstuvo galia: 49 W

Fiderio slopinimas: 1,39 dB

Antenos aukštis: 4,6 m

Matavimo aukštis: 1,6 m

Galia prie antenos: 38,5 W

Matavimo Eil. Nr.	Atstumas nuo antenos, m	Veikiant siųstuvui		Neveikiant siųstuvui	
		Išmatuota vertė	Matavimo vienetas	Išmatuota vertė	Matavimo vienetas
1	2	3	4	5	6
1	1	0,1719	W/m ²	0,0002	W/m ²
2	2	0,0722	W/m ²	0,0001	W/m ²
3	4	0,0379	W/m ²	0,0001	W/m ²
4	6	0,0638	W/m ²	0,0002	W/m ²
5	8	0,0345	W/m ²	0,0001	W/m ²
6	10	0,0697	W/m ²	0,0001	W/m ²
7	15	0,0967	W/m ²	0,0002	W/m ²
8	20	0,0616	W/m ²	0,0001	W/m ²
9	25	0,0585	W/m ²	0,0001	W/m ²
10	30	0,0987	W/m ²	0,0002	W/m ²
11	35	0,1715	W/m ²	0,0001	W/m ²
12	40	0,1893	W/m ²	0,0002	W/m ²
13	45	0,1234	W/m ²	0,0002	W/m ²
14	50	0,1083	W/m ²	0,0001	W/m ²
15	60	0,0573	W/m ²	0,0002	W/m ²
16	70	0,0415	W/m ²	0,0002	W/m ²
17	80	0,0253	W/m ²	0,0001	W/m ²
18	90	0,0225	W/m ²	0,0001	W/m ²
19	100	0,0126	W/m ²	0,0001	W/m ²
20	110	0,0074	W/m ²	0,0002	W/m ²
21	120	0,0066	W/m ²	0,0002	W/m ²
22	130	0,0053	W/m ²	0,0002	W/m ²
23	140	0,0035	W/m ²	0,0002	W/m ²
24	150	0,0047	W/m ²	0,0003	W/m ²
25	160	0,0051	W/m ²	0,0003	W/m ²
26	170	0,0069	W/m ²	0,0001	W/m ²
27	180	0,0091	W/m ²	0,0003	W/m ²
28	190	0,0095	W/m ²	0,0003	W/m ²
29	200	0,0057	W/m ²	0,0004	W/m ²
30	220	0,0045	W/m ²	0,0003	W/m ²
31	240	0,0027	W/m ²	0,0004	W/m ²
32	260	0,0015	W/m ²	0,0003	W/m ²
33	280	0,0011	W/m ²	0,0001	W/m ²
34	300	0,0012	W/m ²	0,0002	W/m ²
35	350	0,0004	W/m ²	0,0002	W/m ²
36	400	0,0001	W/m ²	0,0001	W/m ²

Matavimą atliko: Saulius Dreinius

C PRIEDAS. PROGRAMOS IŠEITIS KODAS

```
public partial class MainForm : Form
{
    private static System.Drawing.Size
    mouseOffset;
    static public int tX = 5, tY = 5, pX, pY, mZ
    = 2, mL = 0, mX2, mY2, dX, dY, cX, cY, sX,
    sY;
    static bool mmove = false, tileUpdate =
    false, tilesShow = false;

    static double[] resol = new double[12] {
    1587.50317500635, 793.751587503175,
    529.167725002117, 264.583862501058,
    132.291931250529, 52.9167725002117,
    26.4583862501058, 13.2291931250529,
    6.61459656252646, 2.64583862501058,
    1.32291931250529, 0.529167725002117 };
    static string[] layers = new string[3] {
    "mapslt", "mapslt ortofoto",
    "mapslt_ortofoto_2005" };
    static string[] layerSuf = new string[3] {
    "png", "jpg", "jpg" };
    static string[] layerRights = new string[] {
    "Hnit-Baltic žemėlapis, GDB10LT duomenys",
    "NŽT prie ŽŪM, ORT10LT 2010 m.", "NŽT prie
    ŽŪM, ORT10LT 2005 m." };
    static bool matavimas = false;
    static int originx = -5122000;
    static int originy = 10000100;

    string roFailas = "";
    static string[] tiles = new string[100];
    static int[] tilesX = new int[100];
    static int[] tilesY = new int[100];
    static string[] tileN = new string[0];
    static Image[] tileD = new Image[0];
    static double[,] spalvos = new double[,] { {
    100, 0, 0, 2 }, { 150, 0, 0, 4 }, { 200, 0,
    0, 7 }, { 255, 0, 0, 10 }, { 0, 255, 100, 15
    } };
    static int spalvuPermatomumas = 50;
    static bool apskaiciuta = false;
    public static bool _apskaiciuta
    {
        get { return apskaiciuta; }
        set { apskaiciuta = value; }
    }
    public static double[,] _spalvos
    {
        get { return spalvos; }
        set { spalvos = value; }
    }
    public static int _spalvuPermatomumas
    {
        get { return spalvuPermatomumas; }
        set { spalvuPermatomumas = value; }
    }
    static double matavimoAukstis=1.5;
    static int centrineBaze = 0;
    public static int _centrineBaze
    {
        get { return centrineBaze; }
        set { centrineBaze = value; }
    }
    public static double _matavimoAukstis
    {
        get { return matavimoAukstis; }
        set { matavimoAukstis = value; }
    }

    public static string direktorija =
    Path.GetDirectoryName(Application.ExecutableP
    ath);
```

```
public static string _direktorija
{
    get { return direktorija; }
    set { direktorija = value; }
}
private static bool tikslinimas;
public static bool _tikslinimas
{
    get { return tikslinimas; }
    set { tikslinimas = value; }
}
private static int mX = 497086, mY = 6111333;
public static int _mX
{
    get { return mX; }
    set { mX = value; }
}
public static int _mY
{
    get { return mY; }
    set { mY = value; }
}
public static int _cX
{
    get { return cX; }
    set { cX = value; }
}
public static int _cY
{
    get { return cY; }
    set { cY = value; }
}
public static int _mZ
{
    get { return mZ; }
    set { mZ = value; }
}

private static double Kr = 1, Kh = 1.5;
private static int Wgs = 2, Spind = 600;
private static bool greiciau = true;
public static double _Kr
{
    get { return Kr; }
    set { Kr = value; }
}
public static double _Kh
{
    get { return Kh; }
    set { Kh = value; }
}
public static int _Wgs
{
    get { return Wgs; }
    set { Wgs = value; }
}
public static int _Spind
{
    get { return Spind; }
    set { Spind = value; }
}
public static bool _greiciau
{
    get { return greiciau; }
    set { greiciau = value; }
}
skaiciavimas_skaiciavimas = new
skaiciavimas();
stotys_stotys = new stotys();
centravimas_centravimas = new centravimas();
paleidimas_paleidimas = new paleidimas();
Form2_nustatymai = new Form2();
```

```

    eksportavimas _eksportavimas = new
    eksportavimas();
    simuliavimas _simuliavimas = new
    simuliavimas();
    minimizavimas _minimizavimas = new
    minimizavimas();
    public MainForm()
    {
        MainForm.panell1 = new
        EM_spinduliute.MainForm.DoubleBufferPanel();
        MainForm.panell1.BackColor =
        System.Drawing.Color.White;
        MainForm.panell1.Location = new
        System.Drawing.Point(0, 0);
        MainForm.panell1.Name = "panell1";
        MainForm.panell1.Size = new
        System.Drawing.Size(640, 480);
        MainForm.panell1.TabIndex = 8;
        MainForm.panell1.Paint += new
        System.Windows.Forms.PaintEventHandler(this.p
        anell1_Paint);
        InitializeComponent();
        this.Controls.Add(MainForm.panell1);

        uzkraukNustatymus();

        if (!Directory.Exists(@direktorija +
        "\\zemelapis"))
        {
            System.IO.Directory.CreateDirectory(direktori
            ja + "\\zemelapis");
        }
        if (!Directory.Exists(@MainForm._direktorija
        + "\\antenos"))
        {
            System.IO.Directory.CreateDirectory(MainForm.
            _direktorija + "\\antenos");
        }
        if (!Directory.Exists(@direktorija +
        "\\objektai"))
        {
            System.IO.Directory.CreateDirectory(direktori
            ja + "\\objektai");
        }
        if (!Directory.Exists(@direktorija +
        "\\skaiciavimai"))
        {
            System.IO.Directory.CreateDirectory(direktori
            ja + "\\skaiciavimai");
        }
        drob(true);
        ServicePointManager.DefaultConnectionLimit =
        5;
        AppDomain.CurrentDomain.ProcessExit += new
        EventHandler(OnProcessExit);
        panell1.MouseMove += new
        System.Windows.Forms.MouseEventHandler(this.c
        ontrol_MouseMove);
        panell1.MouseDown += new
        System.Windows.Forms.MouseEventHandler(this.c
        ontrol_MouseDown);
        panell1.MouseUp += new
        System.Windows.Forms.MouseEventHandler(this.c
        ontrol_MouseUp);
        this.MouseWheel += new
        System.Windows.Forms.MouseEventHandler(panell1
        _MouseWheel);
        System.Windows.Forms.Timer timer = new
        System.Windows.Forms.Timer();
        timer.Tick += new EventHandler(timer_Tick);
        timer.Interval = 400;
        timer.Enabled = true;
        timer.Start();

```

```

        _nustatymai.rodyk();

        _skaiciavimas.nuskaitykBazes(MainForm._direktor
        ija+"\\objektai\\ok.txt");
        _stotys.rodyk();
        _stotys.pirmasAtidarymas();
        toolStripStatusLabel1.Text = "sąrašas dar
        neišsaugotas";
    }
    void timer_Tick(object sender, EventArgs e)
    {
        if (tileUpdate != false)
        {
            piesk();
            tileUpdate = false;
        }
        private void drob(bool pradzia)
        {
            if (pradzia == true)
            {
                Rectangle resolution =
                Screen.PrimaryScreen.Bounds;
                this.ClientSize = new
                System.Drawing.Size((resolution.Width / 100) *
                80, (resolution.Height / 100) * 80);
                toolStripStatusLabel5.Text = layerRights[mL];
            }
            panell1.Width = this.ClientRectangle.Width;
            panell1.Height = this.ClientRectangle.Height;
            tX =
            (int)(Math.Ceiling(this.ClientRectangle.Width *
            1.1 / 256.0)) + 1;
            tY =
            (int)(Math.Ceiling(this.ClientRectangle.Height
            * 1.1 / 256.0)) + 1;
            Array.Resize(ref tiles, tX * tY);
            Array.Resize(ref tilesX, tX * tY);
            Array.Resize(ref tilesY, tX * tY);
            piesk();
        }

        private static string getTile(int[] ret)
        {
            string tileX = ret[0].ToString("X").ToLower();
            string tileY = ret[1].ToString("X").ToLower();
            string saveDir = direktoriija + "\\zemelapis\\"
            + layers[mL] + "\\" + mZ + "\\" + tileY + "";
            string saveFile = saveDir + "\\" + tileX + "."
            + layerSuf[mL];
            string linkFile = "http://dc1.maps.lt/cache/"
            + layers[mL] + "/map/_alllayers/L" +
            mZ.ToString().PadLeft(2, '0') + "/R0000" +
            tileY.PadLeft(4, '0') + "/C0000" +
            tileX.PadLeft(4, '0') + "." + layerSuf[mL];
            FileInfo f = new FileInfo(saveFile);
            if (!f.Exists || f.Length == 0)
            {
                System.IO.Directory.CreateDirectory(saveDir);

                WebClient Client = new WebClient();
                try
                {
                    Client.DownloadFileCompleted += new
                    AsyncCompletedEventHandler(getTileCompleted);
                    Client.DownloadFileAsync(new Uri(linkFile),
                    @saveFile);
                }
                catch { }

                /* Thread newThread = new
                Thread(MainForm.DoWork);
                newThread.Start(saveFile + "|" + saveDir + "|"
                + linkFile + "|" + saveFile); */
            }
            return saveFile;
        }
    }

```

```

    public static void getTileCompleted(object
sender, AsyncCompletedEventArgs e)
    {
        if (mmove != true)
        {
            tileUpdate = true;
        }
    }
    private void panell1_Paint(object sender,
PaintEventArgs e)
    {
        if (tilesShow == true)
        {
            for (int i = 0; i < tiles.Length; i++)
            {
                int tmp = Array.IndexOf(tileN, tiles[i]);
                if (tmp >= 0)
                {
                    if (mmove == true && greiciau == true)
                    {
                        e.Graphics.DrawImage(tileD[tmp], tilesX[i] +
pX, tilesY[i] + pY, 256, 256);
                    }
                    else
                    {
                        e.Graphics.DrawImage(tileD[tmp], tilesX[i],
tilesY[i], 256, 256);
                    }
                }
            }

            if (mZ == 11)
            {
                e.Graphics.InterpolationMode =
System.Drawing.Drawing2D.InterpolationMode.Ne
arestNeighbor;
            }
            else
            {
                e.Graphics.InterpolationMode =
System.Drawing.Drawing2D.InterpolationMode.De
fault;
            }
            if (apskaiciuta == true && mmove != true &&
File.Exists(Path.GetDirectoryName(Application
.ExecutablePath) +
"\\skaiciavimai\\suminis.png"))
            {
                int startX =
(int)Math.Round(Convert.ToDouble(skaiciavimas
._bazes[centrineBaze, 10])) - Spind;
                int startY =
(int)Math.Round(Convert.ToDouble(skaiciavimas
._bazes[centrineBaze, 11])) + Spind;
                int a = (int)Math.Round((startX - mX) /
resol[mZ]);
                int b = (int)Math.Round((startY - mY) /
resol[mZ]);
                int picW = (int)Math.Round(Spind*2 /
resol[mZ]);
                int picH = (int)Math.Round(Spind*2 /
resol[mZ]);
                Image newImage =
Image.FromFile(Path.GetDirectoryName(Applicat
ion.ExecutablePath) +
"\\skaiciavimai\\suminis.png");
                e.Graphics.DrawImage(newImage, (panell.Width
/ 2) + a, (panell.Height / 2) - b, picW,
picH);
                newImage.Dispose();
            }
            if (cX > 0 && mmove != true)
            {
                int a = (int)Math.Round((cX - mX) /
resol[mZ]);

```

```

                int b = (int)Math.Round((cY - mY) /
resol[mZ]);
                if (Math.Abs(a) > (panell.Width / 2) + 10 ||
Math.Abs(b) > (panell.Height / 2) + 10) { }
                else
                {
                    System.Drawing.Pen myPen3 = new
System.Drawing.Pen(System.Drawing.Color.Blue);
                    e.Graphics.DrawLine(myPen3, (panell.Width / 2)
+ a - 6, (panell.Height / 2) - b, (panell.Width
/ 2) + a + 6, (panell.Height / 2) - b);
                    e.Graphics.DrawLine(myPen3, (panell.Width / 2)
+ a, (panell.Height / 2) - b - 6, (panell.Width
/ 2) + a, (panell.Height / 2) - b + 6);
                    myPen3.Dispose();
                }
            }
            if (matavimas == true && mmove != true && cX
!= 0)
            {
                int a2 = (int)Math.Round((cX - mX) /
resol[mZ]);
                int b2 = (int)Math.Round((cY - mY) /
resol[mZ]);

                int a = (int)Math.Round((sX - mX) /
resol[mZ]);
                int b = (int)Math.Round((sY - mY) /
resol[mZ]);
                if (Math.Abs(a) > (panell.Width / 2) + 10 ||
Math.Abs(b) > (panell.Height / 2) + 10) { }
                else
                {
                    System.Drawing.Pen myPen3 = new
System.Drawing.Pen(System.Drawing.Color.Blue);
                    e.Graphics.DrawLine(myPen3, (panell.Width / 2)
+ a - 6, (panell.Height / 2) - b, (panell.Width
/ 2) + a + 6, (panell.Height / 2) - b);
                    e.Graphics.DrawLine(myPen3, (panell.Width / 2)
+ a, (panell.Height / 2) - b - 6, (panell.Width
/ 2) + a, (panell.Height / 2) - b + 6);
                    myPen3.Dispose();
                }
                double theta = Math.Atan2(a-a2, b-b2);
                if (theta < 0)
                {
                    theta += 2 * Math.PI;
                }
                theta = theta * 180 / Math.PI;
                toolStripStatusLabel3.Text =
Math.Round(Math.Sqrt(Math.Pow(a - a2, 2) +
Math.Pow(b - b2, 2)) * resol[mZ]).ToString() +
" m, " + Math.Round(theta).ToString() + "°";
                System.Drawing.Pen myPen;
                myPen = new
System.Drawing.Pen(System.Drawing.Color.Red);
                e.Graphics.DrawLine(myPen, (panell.Width / 2)
+ a, (panell.Height / 2) - b, (panell.Width /
2) + a2, (panell.Height / 2) - b2);
                myPen.Dispose();
            }
        }
    }
    public class DoubleBufferPanel : Panel
    {
        public DoubleBufferPanel()
        {
            this.DoubleBuffered = true;
        }
    }
    public static void UpdateScreen()
    {
        if (panell.InvokeRequired == true)
        {
            panell.Invoke(new
MethodInvoker(UpdateScreen));
        }
    }

```

```

else
{
    panell.Refresh();
}
}
public static void uzdaryk()
{
    piesk();
}
public static void centruok(double[] k)
{
    cX = mX = (int)Math.Round(k[0]);
    cY = mY = (int)Math.Round(k[1]);
    matavimas = false;
    piesk();
}
private void control_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == System.Windows.Forms.MouseButtons.Left)
    {
        mX2 = mX;
        mY2 = mY;
        mouseOffset = new System.Drawing.Size(e.Location);
        mmmove = true;
    }
    if (e.Button == System.Windows.Forms.MouseButtons.Right)
    {
        if ((Control.ModifierKeys & Keys.Shift) == Keys.Shift && cX != 0)
        {
            System.Drawing.Point newLocationOffset = e.Location - mouseOffset;
            mouseCords(e, true);
            sX = dX;
            sY = dY;
            matavimas = true;
        }
        else
        {
            matavimas = false;
            mouseCords(e, true);
            cX = dX;
            cY = dY;
            mouseOffset = new System.Drawing.Size(e.Location);
            NumberFormatInfo nfi = new CultureInfo("en-US", false).NumberFormat;
            nfi.NumberDecimalSeparator = ".";
            if (Wgs == 0)
            {
                toolStripStatusLabel3.Text = "LKS:" + cX + ", " + cY;
            }
            else if (Wgs == 2)
            {
                double[] t = wgslks.lks2wgs(new double[] { cX, cY });
                toolStripStatusLabel3.Text = "WGS: " + Math.Round(t[0], 6).ToString(nfi) + ", " + Math.Round(t[1], 6).ToString(nfi);
            }
            int tempX = Spind + cX - Convert.ToInt32(skaiciavimas._bases[centrineBaze, 10]);
            int tempY = Spind + Convert.ToInt32(skaiciavimas._bases[centrineBaze, 11]) - cY;
            if (tempX < Spind*2 && tempY < Spind*2 && tempX >= 0 && tempY >= 0 && apskaiciuta == true)
            {
                toolStripStatusLabel3.Text += " - " + skaiciavimas._suminis[tempX, tempY].ToString() + "  $\mu\text{W}/\text{cm}^2$ ";
            }
        }
    }
    private void mouseCords(MouseEventArgs e, bool right)
    {
        int t = 0; if (right == true) { t = 0; }
        dX = mX - (int)Math.Round((panell.Width / 2 - e.Location.X + t) * resol[mZ]);
        dY = mY + (int)Math.Round((panell.Height / 2 - e.Location.Y - t) * resol[mZ]);
    }
    private void control_MouseUp(object sender, MouseEventArgs e)
    {
        mmmove = false;
        pX = 0;
        pY = 0;
        piesk();
    }
    private static void apsivalyk()
    {
        for (int i = 0; i < tileD.Length; i++)
        {
            tileD[i].Dispose();
        }
        Array.Clear(tileD, 0, tileD.Length);
        Array.Clear(tileN, 0, tileN.Length);
        Array.Resize(ref tileD, 0);
        Array.Resize(ref tileN, 0);
    }
    public static int[] calcTile(int x, int y)
    {
        int[] ret = new int[4];
        double originX = 256 * resol[mZ] + originx;
        ret[0] = (int)Math.Ceiling((x - originX) / 256 / resol[mZ]);
        double originY = originy - 256 * resol[mZ];
        ret[1] = (int)Math.Ceiling((originY - y) / 256 / resol[mZ]);
        ret[2] = (int)Math.Round(((ret[0] - 1) * 256 * resol[mZ]) + originX);
        ret[3] = (int)Math.Round(originY - ((ret[1]) * 256 * resol[mZ]));
        return ret;
    }
    public static void piesk()
    {
        if (greiciau == true)
        {
            apsivalyk();
        }
        else
        {
            if (tileD.Length > 200)
            {
                apsivalyk();
            }
            double panW = (panell.Width * resol[mZ]) / 2;
            double panH = (panell.Height * resol[mZ]) / 2;
            int[] ret = calcTile((int)Math.Round(mX - panW), (int)Math.Round(mY - panH));
            Image newImage;
            int i = 0;
            for (int fy = 0; fy < tY; fy++)
            {
                for (int fx = 0; fx < tX; fx++)

```

```

    {
        tiles[i] = getTile(new int[] { (int)ret[0] +
fx, (int)ret[1] - fy });
        tilesX[i] = (256 * fx) - (int)Math.Round((mX
- ret[2]) / resol[mZ]) + (panell.Width / 2);
        tilesY[i] = (256 * fy * -1) + (panell.Height
- 256) + (int)Math.Round((mY - ret[3]) /
resol[mZ]) - (panell.Height / 2);
        if (!tileN.Contains(tiles[i]))
        {
            try
            {
                newImage = Image.FromFile(tiles[i]);
                Array.Resize(ref tiled, tiled.Length + 1);
                tiled[tiled.Length - 1] = newImage;

                Array.Resize(ref tileN, tileN.Length + 1);
                tileN[tileN.Length - 1] = tiles[i];
            }
            catch { }
        }
        i++;
    }
    tilesShow = true;
    UpdateScreen();
}

public void control_MouseMove(object sender,
MouseEventArgs e)
{
    if (mmove == true)
    {
        System.Drawing.Point newLocationOffset =
e.Location - mouseOffset;
        mX = (int)Math.Round(mX2 -
(newLocationOffset.X * resol[mZ]));
        mY = (int)Math.Round(mY2 +
(newLocationOffset.Y * resol[mZ]));

        if (greiciau == true)
        {
            pX = newLocationOffset.X;
            pY = newLocationOffset.Y;
            UpdateScreen();
        }
        else
        {
            piesk();
        }
    }
    private void toolStripButton7_Click(object
sender, EventArgs e)
    {
        _nustatymai.ShowDialog();
    }
    private void Form_Resize(object sender,
EventArgs e)
    {
        if (WindowState ==
FormWindowState.Maximized)
        {
            drobe(false);
        }
        if (WindowState == FormWindowState.Normal)
        {
            drobe(false);
        }
    }
    private void panell1_MouseWheel(object
sender, MouseEventArgs e)
    {
        mouseCords(e, false);
        if (e.Delta > 0 && mZ < 11)
        {

```

```

            mX = dX - (int)Math.Round((e.Location.X -
panell1.Width / 2) * resol[mZ + 1]);
            mY = dY - (int)Math.Round((panell1.Height / 2 -
e.Location.Y) * resol[mZ + 1]);
            zoomIn();
        }
        else if (e.Delta < 0 && mZ > 1)
        {
            mX = dX - (int)Math.Round((e.Location.X -
panell1.Width / 2) * resol[mZ - 1]);
            mY = dY - (int)Math.Round((panell1.Height / 2 -
e.Location.Y) * resol[mZ - 1]);
            zoomOut();
        }
    }
    private void toolStripButton10_Click(object
sender, EventArgs e)
    {
        zoomOut();
    }
    private void toolStripButton9_Click(object
sender, EventArgs e)
    {
        zoomIn();
    }
    private void zoomOut()
    {
        if (mZ > 1)
        {
            mZ--;
            piesk();
        }
        apsivalyk();
    }
    private void zoomIn()
    {
        if (mZ < 11)
        {
            mZ++;
            piesk();
        }
        apsivalyk();
    }
    private void
vektorinisToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        mL = 0;
        piesk();
        toolStripStatusLabel5.Text = layerRights[mL];
    }
    private void
orotofoto20120ToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        mL = 1;
        piesk();
        toolStripStatusLabel5.Text = layerRights[mL];
    }
    private void
ortofoto2005ToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        mL = 2;
        piesk();
        toolStripStatusLabel5.Text = layerRights[mL];
    }
    private void toolStripButton5_Click(object
sender, EventArgs e)
    {
        mL = 0;
        piesk();
        toolStripStatusLabel5.Text = layerRights[mL];
    }
    private void toolStripButton4_Click(object
sender, EventArgs e)
    {

```

```

        mL = 1;
        piesk();
        toolStripStatusLabel5.Text =
layerRights[mL];
    }
    private void toolStripButton6_Click(object
sender, EventArgs e)
    {
        mL = 2;
        piesk();
        toolStripStatusLabel5.Text =
layerRights[mL];
    }
    private void toolStripButton8_Click(object
sender, EventArgs e)
    {
        _paleidimas.rodyk();
        _paleidimas.ShowDialog();
    }

    private void toolStripButton11_Click(object
sender, EventArgs e)
    {
        mX = 497086; mY = 6111333; mZ = 2; piesk();
    }
    private Form1 _Form1 = new Form1();
    private void toolStripButton13_Click(object
sender, EventArgs e)
    {
        _Form1.ShowDialog();
    }

    private void toolStripButton14_Click(object
sender, EventArgs e)
    {
        if (tikslinimas == true)
        {
            _stotys.tikslinimas(new double[] { cX, cY
});
        }
        else
        {
            _stotys.uzpildyk(stotys._listViewSkaicius);
        }
        _stotys.ShowDialog();
    }
    private void toolStripButton15_Click(object
sender, EventArgs e)
    {
        _centravimas.rodyk();
        _centravimas.ShowDialog();
    }
    private void MainForm_ParentChanged(object
sender, EventArgs e)
    {
        MessageBox.Show(" ");
    }
    private void stumk(char kur)
    {
        if (kur == 'l')
        {
            mX = (int)Math.Round(mX - (150 *
resol[mZ]));
        }
        else if (kur == 'r')
        {
            mX = (int)Math.Round(mX + (150 *
resol[mZ]));
        }
        else if (kur == 'u')
        {
            mY = (int)Math.Round(mY + (150 *
resol[mZ]));
        }
        else if (kur == 'd')
        {

```

```

            mY = (int)Math.Round(mY - (150 * resol[mZ]));
        }
        piesk();
    }
    private void MainForm_KeyDown(object sender,
System.Windows.Forms.KeyEventArgs e)
    {
        switch (e.KeyCode)
        {
            case Keys.Left:
                stumk('l');
                break;
            case Keys.Right:
                stumk('r');
                break;
            case Keys.Up:
                stumk('u');
                break;
            case Keys.Down:
                stumk('d');
                break;
            default:
                return;
        }
    }
    private void toolStripButton2_Click(object
sender, EventArgs e)
    {
        OpenFileDialog openFileDialog1 = new
OpenFileDialog();
        openFileDialog1.InitialDirectory =
Path.GetDirectoryName(direktorija+"\\objektai\\
");
        openFileDialog1.Filter = "Radiotechninių
objektų sąrašas (*.txt)|*.txt";
        openFileDialog1.Title = "Pasirinkite
radiotechninių objektų sąrašo failą";
        if (openFileDialog1.ShowDialog() ==
DialogResult.OK)
        {
            roFailas = openFileDialog1.FileName;
            _skaiciavimas.nuskaitykBazes(roFailas);
            _stotys.rodyk();
            _stotys.pirmasAtidarymas();
            apskaiciuta = false;
        }
        toolStripStatusLabel1.Text =
Path.GetFileName(roFailas).ToString();
    }
    private void irasyk()
    {
        StreamWriter file = new
StreamWriter(roFailas);
        string tmp = "";
        for (int i = 0; i <
skaiciavimas._bazes.GetLength(0); i++)
        {
            tmp = skaiciavimas._bazes[i, 0] + "|" +
skaiciavimas._bazes[i, 1] + "|" +
skaiciavimas._bazes[i, 2] + "|" +
skaiciavimas._bazes[i, 3] + "|" +
skaiciavimas._bazes[i, 4] + "|" +
skaiciavimas._bazes[i, 5] + "|" +
skaiciavimas._bazes[i, 6] + "|" +
skaiciavimas._bazes[i, 7] + "|" +
skaiciavimas._bazes[i, 8] + "|" +
skaiciavimas._bazes[i, 9] + "|" +
skaiciavimas._bazes[i, 10] + "|" +
skaiciavimas._bazes[i, 11];
            file.WriteLine(tmp);
        }
        file.Close();
        toolStripStatusLabel1.Text =
Path.GetFileName(roFailas).ToString();
    }

```

```

private void toolStripButton3_Click(object sender, EventArgs e)
{
    if (!File.Exists(roFailas) ||
    (Control.ModifierKeys & Keys.Shift) ==
    Keys.Shift)
    {
        SaveFileDialog saveFileDialog1 = new
        SaveFileDialog();
        saveFileDialog1.InitialDirectory =
        Path.GetDirectoryName(direktorija +
        "\\objektai\\");
        saveFileDialog1.Filter = "Radiotechninių
        objektų sąrašas (*.txt)|*.txt";
        saveFileDialog1.Title = "Išsaugokite
        radiotechninių objektų sąrašo failą";
        saveFileDialog1.ShowDialog();
        if (saveFileDialog1.FileName != "")
        {
            roFailas =
            saveFileDialog1.FileName.ToString();
            irasyk();
        }
        else
        {
            irasyk();
            MessageBox.Show("Radiotechninių objektų
            sąrašas išsaugotas");
        }
    }

    static void OnProcessExit(object sender,
    EventArgs e)
    {
        NumberFormatInfo nfi = new CultureInfo("en-
        US", false).NumberFormat;
        nfi.NumberDecimalSeparator = ".";
        StreamWriter file = new
        StreamWriter(direktorija +
        "\\konfiguracija.txt");
        string nustat = mX.ToString(nfi) + "|" +
        mY.ToString(nfi) + "|" + mZ + "|" + mL + "|" +
        Kr.ToString(nfi) + "|" + Kh.ToString(nfi) +
        "|" + Wgs + "|" +
        matavimoAukstis.ToString(nfi) + "|" +
        Spind.ToString(nfi);
        file.WriteLine(nustat);

        for (int i = 0; i < 5; i++)
        {
            for (int ii = 0; ii < 4; ii++)
            {
                file.Write(spalvos[i, ii].ToString(nfi) +
                "|");
            }
            file.Write(spalvuPermatomumas.ToString());
            file.Close();
        }
        private void uzkraukNustatymus()
        {
            NumberFormatInfo nfi = new CultureInfo("en-
            US", false).NumberFormat;
            nfi.NumberDecimalSeparator = ".";
            FileInfo f = new FileInfo(@direktorija +
            "\\konfiguracija.txt");
            if (f.Exists && f.Length != 0)
            {
                string[] lines =
                System.IO.File.ReadAllLines(direktorija +
                "\\konfiguracija.txt");
                string[] splitas = lines[0].Split('|');
                mX = Convert.ToInt32(splitas[0], nfi);
                mY = Convert.ToInt32(splitas[1], nfi);
                mZ = Convert.ToInt32(splitas[2], nfi);
                mL = Convert.ToInt32(splitas[3], nfi);

                Kr = Convert.ToDouble(splitas[4], nfi);
                Kh = Convert.ToDouble(splitas[5], nfi);
                Wgs = Convert.ToInt32(splitas[6], nfi);
                matavimoAukstis = Convert.ToDouble(splitas[7],
                nfi);
                Spind = Convert.ToInt32(splitas[8], nfi);
                splitas = lines[1].Split('|');
                int n=0;
                for (int i = 0; i < 5; i++)
                {
                    for (int ii = 0; ii < 4; ii++)
                    {
                        spalvos[i, ii] = Convert.ToDouble(splitas[n],
                        nfi);
                        n++;
                    }
                }
                spalvuPermatomumas =
                Convert.ToInt32(splitas[n]);
            }
        }
        private void toolStripButton1_Click(object sender, EventArgs e)
        {
            _skaiciavimas.nuskaitykBazes("pradzia.nera");
            _stotys.rodyk();
            _stotys.pirmasAtidarymas();
            roFailas = "";
            toolStripStatusLabel1.Text = "sąrašas dar
            neišsaugotas";
        }
        private void toolStripButton12_Click(object sender, EventArgs e)
        {
            if (apskaiciuta == true)
            {
                _eksportavimas.rodyk();
                _eksportavimas.ShowDialog();
            }
            else
            {
                MessageBox.Show("Pirmiau atlikite
                skaičiavimą");
            }
        }
        private void toolStripButton16_Click(object sender, EventArgs e)
        {
            if (apskaiciuta == true && stotys._keistas2 ==
            false)
            {
                _simuliavimas.rodyk();
                _simuliavimas.ShowDialog();
            }
            else
            {
                MessageBox.Show("Pirmiau atlikite
                skaičiavimą");
            }
        }
        private void toolStripButton17_Click(object sender, EventArgs e)
        {
            if (minimizavimas._taskaiNr > 0)
            {
                minimizavimas._taskai[minimizavimas._taskaiNr, 0]
                = cX;
                minimizavimas._taskai[minimizavimas._taskaiNr,
                1] = cY;
            }
            _minimizavimas.rodyk();
            _minimizavimas.ShowDialog();
        }
    }
}
public partial class Form2 : Form

```

```

{
public Form2()
{
InitializeComponent();
}
public void rodyk()
{
NumberFormatInfo nfi = new CultureInfo("en-
US", false).NumberFormat;
nfi.NumberDecimalSeparator = ".";
textBox1.Text = MainForm._Kr.ToString(nfi);
textBox2.Text = MainForm._Kh.ToString(nfi);
if (MainForm._Wgs == 0)
{
radioButton1.Checked = true;
}
if (MainForm._Wgs == 1)
{
radioButton2.Checked = true;
}
if (MainForm._Wgs == 2)
{
radioButton3.Checked = true;
}
if (MainForm._greiciau == true)
{
checkBox1.Checked = false;
}
else
{
checkBox1.Checked = true;
}
this.button2.BackColor =
System.Drawing.Color.FromArgb(((int)((byte) (
MainForm._spalvos[0,0]))),
((int)((byte) (MainForm._spalvos[0,1]))),
((int)((byte) (MainForm._spalvos[0,2]))));
this.button3.BackColor =
System.Drawing.Color.FromArgb(((int)((byte) (
MainForm._spalvos[1,0]))),
((int)((byte) (MainForm._spalvos[1,1]))),
((int)((byte) (MainForm._spalvos[1,2]))));
this.button4.BackColor =
System.Drawing.Color.FromArgb(((int)((byte) (
MainForm._spalvos[2,0]))),
((int)((byte) (MainForm._spalvos[2,1]))),
((int)((byte) (MainForm._spalvos[2,2]))));
this.button5.BackColor =
System.Drawing.Color.FromArgb(((int)((byte) (
MainForm._spalvos[3,0]))),
((int)((byte) (MainForm._spalvos[3,1]))),
((int)((byte) (MainForm._spalvos[3,2]))));
this.button6.BackColor =
System.Drawing.Color.FromArgb(((int)((byte) (
MainForm._spalvos[4,0]))),
((int)((byte) (MainForm._spalvos[4,1]))),
((int)((byte) (MainForm._spalvos[4,2]))));
this.textBox3.Text = MainForm._spalvos[0,
3].ToString(nfi);
this.textBox4.Text = MainForm._spalvos[1,
3].ToString(nfi);
this.textBox5.Text = MainForm._spalvos[2,
3].ToString(nfi);
this.textBox6.Text = MainForm._spalvos[3,
3].ToString(nfi);
this.textBox7.Text = MainForm._spalvos[4,
3].ToString(nfi);
this.textBox8.Text =
MainForm._spalvuPermatomumas.ToString();
this.textBox9.Text =
MainForm._Spind.ToString();

}
private void textBox1_KeyPress(object
sender, KeyPressEventArgs e)

```

```

{
if (!char.IsControl(e.KeyChar) &&
!char.IsDigit(e.KeyChar) && e.KeyChar != '.')
{
e.Handled = true;
}
if (e.KeyChar == '.' && (sender as
TextBox).Text.IndexOf('.') > -1)
{
e.Handled = true;
}
}
private void textBox2_KeyPress(object sender,
KeyPressEventArgs e)
{
if (!char.IsControl(e.KeyChar) &&
!char.IsDigit(e.KeyChar))
{
e.Handled = true;
}
if (e.KeyChar == '.' && (sender as
TextBox).Text.IndexOf('.') > -1)
{
e.Handled = true;
}
}
private void button1_Click(object sender,
EventArgs e)
{
NumberFormatInfo nfi = new CultureInfo("en-
US", false).NumberFormat;
nfi.NumberDecimalSeparator = ".";
if (int.Parse(textBox8.Text) < 0 ||
int.Parse(textBox8.Text) > 100)
{
textBox8.Text = "50";
}
if (int.Parse(textBox9.Text) < 0)
{
textBox9.Text = "300";
}
MainForm._Kr = Convert.ToDouble(textBox1.Text,
nfi);
MainForm._Kh = Convert.ToDouble(textBox2.Text,
nfi);

if (radioButton1.Checked == true)
{
MainForm._Wgs = 0;
}
if (radioButton2.Checked == true)
{
MainForm._Wgs = 1;
}
if (radioButton3.Checked == true)
{
MainForm._Wgs = 2;
}
if (checkBox1.Checked == false)
{
MainForm._greiciau = true;
}
else
{
MainForm._greiciau = false;
}
MainForm._spalvuPermatomumas =
Convert.ToInt32(textBox8.Text);
MainForm._Spind =
Convert.ToInt32(textBox9.Text);
MainForm._spalvos[0, 0] =
this.button2.BackColor.R;
MainForm._spalvos[0, 1] =
this.button2.BackColor.G;
MainForm._spalvos[0, 2] =
this.button2.BackColor.B;

```

```

    MainForm._spalvos[0, 3] =
Convert.ToDouble(this.textBox3.Text, nfi);
    MainForm._spalvos[1, 0] =
this.button3.BackColor.R;
    MainForm._spalvos[1, 1] =
this.button3.BackColor.G;
    MainForm._spalvos[1, 2] =
this.button3.BackColor.B;
    MainForm._spalvos[1, 3] =
Convert.ToDouble(this.textBox4.Text, nfi);
    MainForm._spalvos[2, 0] =
this.button4.BackColor.R;
    MainForm._spalvos[2, 1] =
this.button4.BackColor.G;
    MainForm._spalvos[2, 2] =
this.button4.BackColor.B;
    MainForm._spalvos[2, 3] =
Convert.ToDouble(this.textBox5.Text, nfi);
    MainForm._spalvos[3, 0] =
this.button5.BackColor.R;
    MainForm._spalvos[3, 1] =
this.button5.BackColor.G;
    MainForm._spalvos[3, 2] =
this.button5.BackColor.B;
    MainForm._spalvos[3, 3] =
Convert.ToDouble(this.textBox6.Text, nfi);
    MainForm._spalvos[4, 0] =
this.button6.BackColor.R;
    MainForm._spalvos[4, 1] =
this.button6.BackColor.G;
    MainForm._spalvos[4, 2] =
this.button6.BackColor.B;
    MainForm._spalvos[4, 3] =
Convert.ToDouble(this.textBox7.Text, nfi);
    this.Close();
}
private void Form_KeyDown(object sender,
EventArgs e)
{
    if (e.KeyCode == Keys.Escape)
    {
        this.Close();
    }
}
private void nustatykSpalva(int i, object
sender)
{
    var Button = sender as Button;
    System.Windows.Forms.ColorDialog
colorDialog1;
    colorDialog1 = new
System.Windows.Forms.ColorDialog();
    if (colorDialog1.ShowDialog() ==
DialogResult.OK)
    {
        Button.BackColor = colorDialog1.Color;
    }
}
private void button2_Click(object sender,
EventArgs e)
{
    nustatykSpalva(0, sender);
}
private void button3_Click(object sender,
EventArgs e)
{
    nustatykSpalva(2, sender);
}
private void button4_Click(object sender,
EventArgs e)
{
    nustatykSpalva(3, sender);
}
private void button5_Click(object sender,
EventArgs e)
{
    nustatykSpalva(4, sender);
}

```

```

}
private void button6_Click(object sender,
EventArgs e)
{
    nustatykSpalva(5, sender);
}
}

public partial class stotys : Form
{
    private ListViewItem listviewitem;
    string[] fileEntries;
    static int listViewSkaicius;
    skaiciavimas _skaiciavimas = new
skaiciavimas();
    NumberFormatInfo nfi;
    public static int _listViewSkaicius
{
    get { return listViewSkaicius; }
    set { listViewSkaicius = value; }
}
    static bool keistas2;
    public static bool _keistas2
{
    get { return keistas2; }
    set { keistas2 = value; }
}
    public stotys()
    {
        InitializeComponent();
        listView1.View =
System.Windows.Forms.View.Details;
        this.listView1.Columns.Add("");
        this.listView1.Columns.Add("");
        this.listView1.Columns[0].Width = 23;
        this.listView1.Columns[1].Width = 137;
        listView1.CheckBoxes = true;
        nfi = new CultureInfo("en-US",
false).NumberFormat;
        nfi.NumberDecimalSeparator = ".";
    }
    public void tikslinimas(double[] cKords)
    {
        MainForm._tikslinimas = false;
        skaiciavimas._bazes[listViewSkaicius, 10] =
cKords[0].ToString();
        skaiciavimas._bazes[listViewSkaicius, 11] =
cKords[1].ToString();
        uzpildyk(listViewSkaicius);
    }
    public void uzpildyk(int i)
    {
        fileEntries =
Directory.GetFiles(@MainForm._direktorija +
"\antenos");
        listViewSkaicius = i;
        if (MainForm._Wgs == 0)
        {
            textBox4.Text = skaiciavimas._bazes[i, 10] +
", " + skaiciavimas._bazes[i, 11];
        }
        else if (MainForm._Wgs == 2)
        {
            double[] tmp = wgs1ks.1ks2wgs(new double[] {
Convert.ToDouble(skaiciavimas._bazes[i, 10]),
Convert.ToDouble(skaiciavimas._bazes[i, 11])
});
            textBox4.Text = Math.Round(tmp[0],
6).ToString(nfi) + ", " + Math.Round(tmp[1],
6).ToString(nfi);
        }
        textBox9.Text = skaiciavimas._bazes[i,
1].ToString();
        textBox5.Text = skaiciavimas._bazes[i,
2].ToString();
        textBox6.Text = skaiciavimas._bazes[i,
3].ToString();
    }
}

```

```

        textBox7.Text = skaiciavimas._bazes[i,
4].ToString();
        textBox8.Text = skaiciavimas._bazes[i,
5].ToString();
        textBox1.Text = skaiciavimas._bazes[i,
6].ToString();
        textBox2.Text = skaiciavimas._bazes[i,
7].ToString();
        textBox3.Text = skaiciavimas._bazes[i,
8].ToString();
        this.comboBox1.Items.Clear();
        this.comboBox1.Items.Add("");
        if (skaiciavimas._bazes[i, 9] == "")
        {
            this.comboBox1.Text = "";
        }
        foreach (string fileName in fileEntries)
        {

this.comboBox1.Items.Add(Path.GetFileName(fil
eName));
        if (skaiciavimas._bazes[i, 9] ==
Path.GetFileName(fileName))
        {
            this.comboBox1.Text =
Path.GetFileName(fileName);
        }
        }
        private void
comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
        {
            if (this.comboBox1.SelectedItem.ToString()
== "")
            {
                textBox3.Text = "0";
            }
            else
            {
                if (skaiciavimas._bazes[listViewSkaicius,
9].ToString() ==
this.comboBox1.SelectedItem.ToString())
                {
                    return;
                }
                string[] lines =
System.IO.File.ReadAllLines(MainForm._direkto
rija + "\\antenos\\" +
this.comboBox1.SelectedItem.ToString());
                foreach (string line in lines)
                {
                    string[] lineS = line.Split(' ');
                    if (lineS[0] == "GAIN")
                    {
                        textBox3.Text = lineS[1].Trim().ToString();
                    }
                }
            }
            public double[] string2kords(string str)
            {
                if (MainForm._Wgs == 0)
                {
                    string[] tmp; double[] lks = new double[2];
                    tmp = str.Split(',');
                    if (tmp.Length == 2)
                    {
                        lks = new double[] {
Convert.ToDouble(tmp[0], nfi),
Convert.ToDouble(tmp[1], nfi) };
                    }
                    else if (tmp.Length == 3)
                    {
                        tmp = str.Split(' ');
                        if (tmp.Length == 2)
                    {

```

```

                        lks = new double[] {
Convert.ToDouble(tmp[0].ToString(nfi)),
Convert.ToDouble(tmp[1].ToString(nfi)) };
                    }
                }
                return lks;
            }
            else if (MainForm._Wgs == 2)
            {
                string[] tmp; double[] lks = new double[2];
                tmp = str.Split(',');
                if (tmp.Length == 2)
                {
                    lks = new double[] { Convert.ToDouble(tmp[0],
nfi), Convert.ToDouble(tmp[1], nfi) };
                }
                else if (tmp.Length == 3)
                {
                    tmp = str.Split(' ');
                    if (tmp.Length == 2)
                    {
                        lks = new double[] {
Convert.ToDouble(tmp[0].ToString(nfi)),
Convert.ToDouble(tmp[1].ToString(nfi)) };
                    }
                    else if (tmp.Length == 1)
                    {
                        tmp = str.Split(' ');
                        if (tmp.Length == 2)
                        {
                            lks = new double[] { Convert.ToDouble(tmp[0],
nfi), Convert.ToDouble(tmp[1], nfi) };
                        }
                    }
                    return wgs1ks.wgs2lks(lks);
                }
            }
            else
            {
                return new double[2];
            }
        }

        public bool issaugok()
        {
            int i = listViewSkaicius;

            string bX = skaiciavimas._bazes[i, 10], bY =
skaiciavimas._bazes[i, 11];

            bool keistas = false;
            if (skaiciavimas._bazes[i, 1] != textBox9.Text
|| skaiciavimas._bazes[i, 3] != textBox6.Text)
            {
                keistas = true;
            }
            double[] bXY = string2kords(textBox4.Text);
            skaiciavimas._bazes[i, 1] = textBox9.Text;
            skaiciavimas._bazes[i, 2] = textBox5.Text;
            skaiciavimas._bazes[i, 3] = textBox6.Text;
            skaiciavimas._bazes[i, 4] = textBox7.Text;
            skaiciavimas._bazes[i, 5] = textBox8.Text;
            skaiciavimas._bazes[i, 6] = textBox1.Text;
            skaiciavimas._bazes[i, 7] = textBox2.Text;
            skaiciavimas._bazes[i, 8] = textBox3.Text;
            skaiciavimas._bazes[i, 10] =
Math.Round(bXY[0]).ToString();

```

```

    skaiciavimas._bazes[i, 11] =
Math.Round(bXY[1]).ToString();
    skaiciavimas._bazes[i, 9] =
this.comboBox1.SelectedItem.ToString();

    for (i = 0; i <
skaiciavimas._bazes.GetLength(0); i++)
    {

        skaiciavimas._bazes[i, 0] =
(listView1.Items[i].Checked == true) ? "@" :
"";
    }
    keistas2 = true;

    return keistas;
}
public void rodyk()
{
    listView1.Items.Clear();
    for (int i = 0; i <
skaiciavimas._bazes.GetLength(0); i++)
    {
        listviewitem = new ListViewItem("");

listviewitem.SubItems.Add(skaiciavimas._bazes
[i, 1] + ", " + skaiciavimas._bazes[i, 3] +
"");
        this.listView1.Items.Add(listviewitem);
        if (skaiciavimas._bazes[i, 0] == "@")
        {
            listviewitem.Checked = true;
        }
    }
}

private void button2_Click(object sender,
EventArgs e) // sukurk
{
    issaugok();
    string[,] tmp = new string[2, 11];
    skaiciavimas._bazes =
_skaiciavimas.ReDimension(skaiciavimas._bazes
, skaiciavimas._bazes.GetLength(0) + 1,
skaiciavimas._bazes.GetLength(1));
    tmp = _skaiciavimas.ReDimension(tmp,
skaiciavimas._bazes.GetLength(0),
skaiciavimas._bazes.GetLength(1));
    for (int i = 0; i <
skaiciavimas._bazes.GetLength(0)-1; i++)
    {
        if (i < listViewSkaicius)
        {
            for (int c = 0; c <
skaiciavimas._bazes.GetLength(1); c++)
            {
                tmp[i, c] = skaiciavimas._bazes[i, c];
            }
        }
        else
        {
            if (i == listViewSkaicius)
            {
                for (int c = 0; c <
skaiciavimas._bazes.GetLength(1); c++)
                {
                    tmp[i, c] = skaiciavimas._bazes[i, c];
                }
            }
            for (int c = 0; c <
skaiciavimas._bazes.GetLength(1); c++)
            {
                tmp[(i + 1), c] = skaiciavimas._bazes[i, c];
            }
        }
        if (listViewSkaicius == i)

```

```

    {
        tmp[(i + 1), 1] = "kopijs " + tmp[i + 1, 1];
    }
}
for (int i = 0; i < tmp.GetLength(0); i++)
{
    for (int c = 0; c < tmp.GetLength(1); c++)
    {
        skaiciavimas._bazes[i, c] = tmp[i, c];
    }
}
rodyk();
uzpildyk(listViewSkaicius+1);
listView1.Items[listViewSkaicius].Selected =
true;
}
private void button3_Click(object sender,
EventArgs e) //istrink
{
    if (skaiciavimas._bazes.GetLength(0) == 1)
        return;
    issaugok();
    string[,] tmp = new string[2, 11];
    tmp = _skaiciavimas.ReDimension(tmp,
skaiciavimas._bazes.GetLength(0) - 1,
skaiciavimas._bazes.GetLength(1));

    for (int i = 0; i <
skaiciavimas._bazes.GetLength(0); i++)
    {
        if (i < listViewSkaicius)
        {
            for (int c = 0; c <
skaiciavimas._bazes.GetLength(1); c++)
            {
                tmp[i, c] = skaiciavimas._bazes[i, c];
            }
        }
        else if (i != listViewSkaicius)
        {
            for (int c = 0; c <
skaiciavimas._bazes.GetLength(1); c++)
            {
                tmp[i - 1, c] = skaiciavimas._bazes[i, c];
            }
        }
        skaiciavimas._bazes =
_skaiciavimas.ReDimension(skaiciavimas._bazes,
skaiciavimas._bazes.GetLength(0) - 1,
skaiciavimas._bazes.GetLength(1));
        for (int i = 0; i <
skaiciavimas._bazes.GetLength(0); i++)
        {
            for (int c = 0; c <
skaiciavimas._bazes.GetLength(1); c++)
            {
                skaiciavimas._bazes[i, c] = tmp[i, c];
            }
        }
        rodyk();
        if (listViewSkaicius ==
skaiciavimas._bazes.GetLength(0)) {
            uzpildyk(listViewSkaicius - 1); } else
            {uzpildyk(listViewSkaicius); }
        listView1.Items[listViewSkaicius].Selected =
true;
    }

    private void button4_Click(object sender,
EventArgs e)
    {
        MainForm._tikslinimas = true;
        bool keistas = issaugok();
        if (keistas == true)
        {

```

```

        rodyk();
    }
    MainForm.centruok(new double[] {
    Convert.ToDouble(skaiciavimas._bazes[listView
    Skaicius, 10], nfi),
    Convert.ToDouble(skaiciavimas._bazes[listView
    Skaicius, 11], nfi) });
    this.Close();

    }

    private void button1_Click(object sender,
    EventArgs e)
    {
        bool keistas = issaugok();
        if (keistas == true)
        {
            rodyk();
        }
        this.Close();
        MainForm._cX = 0;
    }
    private void listView1_MouseUp(object
    sender, MouseEventArgs e)
    {
        bool keistas = issaugok();
        for (int i = 0; i < listView1.Items.Count;
        i++)
        {
            if (listView1.Items[i].Selected == true)
            {
                uzpildyk(i);
            }
        }
        if (keistas == true)
        {
            rodyk();
            listView1.Items[listViewSkaicius].Selected =
            true;
        }
    }
    private void textBox_KeyPress(object sender,
    KeyPressEventArgs e)
    {
        if (!char.IsControl(e.KeyChar) &&
        !char.IsDigit(e.KeyChar) && e.KeyChar != '.')
        {
            e.Handled = true;
        }
        if (e.KeyChar == '.' && (sender as
        TextBox).Text.IndexOf('.') > -1)
        {
            e.Handled = true;
        }
    }
    private void textBox_KeyDown(object sender,
    KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            issaugok();
            rodyk();
        }
    }
    private void textBox_LetterPress(object
    sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == '|')
        {
            e.Handled = true;
        }
    }
    public void pirmasAtidarymas()
    {
        uzpildyk(0);
        listView1.Items[0].Selected = true;
    }

```

```

        private void
        AdjustWidthComboBox_DropDown(object sender,
        System.EventArgs e)
        {
            ComboBox senderComboBox = (ComboBox)sender;
            int width = senderComboBox.DropDownWidth;
            Graphics g = senderComboBox.CreateGraphics();
            Font font = senderComboBox.Font;
            int vertScrollBarWidth =
            (senderComboBox.Items.Count >
            senderComboBox.MaxDropDownItems)
            ? SystemInformation.VerticalScrollBarWidth :
            0;
            int newWidth;
            foreach (string s in ((ComboBox)sender).Items)
            {
                newWidth = (int)g.MeasureString(s, font).Width
                + vertScrollBarWidth;
                if (width < newWidth)
                {
                    width = newWidth;
                }
            }
            senderComboBox.DropDownWidth = width;
        }
        private void Form_KeyDown(object sender,
        KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Escape)
            {
                this.Close();
            }
        }

        public partial class paleidimas : Form
        {
            skaiciavimas _skaiciavimas = new
            skaiciavimas();
            public paleidimas()
            {
                InitializeComponent();
            }
            public void rodyk()
            {
                NumberFormatInfo nfi = new CultureInfo("en-
                US", false).NumberFormat;
                nfi.NumberDecimalSeparator = ".";
                this.comboBox1.Items.Clear();
                for (int i = 0; i <
                skaiciavimas._bazes.GetLength(0); i++)
                {
                    string tt = "";
                    if (MainForm._Wgs == 0)
                    {
                        tt = Convert.ToDouble(skaiciavimas._bazes[i,
                        10], nfi) + ", " +
                        Convert.ToDouble(skaiciavimas._bazes[i, 11],
                        nfi);
                    }
                    else if (MainForm._Wgs == 2)
                    {
                        double[] t = wgs1ks.1ks2wgs(new double[] {
                        Convert.ToDouble(skaiciavimas._bazes[i, 10],
                        nfi), Convert.ToDouble(skaiciavimas._bazes[i,
                        11], nfi) });
                        tt = Math.Round(t[0], 6).ToString(nfi) + ", "
                        + Math.Round(t[1], 6).ToString(nfi);
                    }
                }
                this.comboBox1.Items.Add(skaiciavimas._bazes[i,
                1] + ", " + skaiciavimas._bazes[i, 3] + "°, " +
                tt);
            }
            this.comboBox1.SelectedIndex =
            MainForm._centrineBaze;

```

```

        this.textBox1.Text =
MainForm._matavimoAukstis.ToString(nfi);
    }
    private void textBox_KeyPress(object sender,
KeyPressEventArgs e)
    {
        if (!char.IsControl(e.KeyChar) &&
!char.IsDigit(e.KeyChar) && e.KeyChar != '.')
        {
            e.Handled = true;
        }
        if (e.KeyChar == '.' && (sender as
TextBox).Text.IndexOf('.') > -1)
        {
            e.Handled = true;
        }
    }
    private void button1_Click(object sender,
EventArgs e)
    {
        NumberFormatInfo nfi = new CultureInfo("en-
US", false).NumberFormat;
        nfi.NumberDecimalSeparator = ".";
        MainForm._centrineBaze =
this.comboBox1.SelectedIndex;
        MainForm._matavimoAukstis =
Convert.ToDouble(this.textBox1.Text, nfi);

        _skaiciavimas.apskaiciuokVisus(MainForm._cent
rineBaze, MainForm._matavimoAukstis);
        this.Close();
        stotys._keistas2 = false;
        MainForm.centruok(new double[] {
Convert.ToDouble(skaiciavimas._bases[MainForm
._centrineBaze, 10], nfi),
Convert.ToDouble(skaiciavimas._bases[MainForm
._centrineBaze, 11], nfi) });
    }

    class skaiciavimas
    {
        static string[,] bases = new string[2, 11];
        static double[,] suminis;
        string failuDirektorija =
MainForm._direktorija + "\\skaiciavimai";
        public static string[,] _bases
        {
            get { return bases; }
            set { bases = value; }
        }
        public static double[,] _suminis
        {
            get { return suminis; }
            set { suminis = value; }
        }
        public void nuskaitykBazes(string tmp)
        {
            FileInfo f = new FileInfo(tmp);
            if (f.Exists && f.Length != 0)
            {
                string[] lines =
System.IO.File.ReadAllLines(@tmp);
                bases = ReDimension(bases, lines.Length,
12);
                int i = 0;
                foreach (string line in lines)
                {
                    string[] str = line.Split('|');
                    bases[i, 0] = str[0].ToString();
                    bases[i, 1] = str[1].ToString();
                    bases[i, 2] = str[2].ToString();
                    bases[i, 3] = str[3].ToString();
                    bases[i, 4] = str[4].ToString();
                    bases[i, 5] = str[5].ToString();
                    bases[i, 6] = str[6].ToString();

                    bases[i, 7] = str[7].ToString();
                    bases[i, 8] = str[8].ToString();
                    bases[i, 9] = str[9].ToString();
                    bases[i, 10] = str[10].ToString();
                    bases[i, 11] = str[11].ToString();
                    i++;
                }
            }
            else
            {
                bases = ReDimension(bases, 1, 12);
                bases[0, 0] = "@";
                bases[0, 1] = "BTS";
                bases[0, 2] = "0";
                bases[0, 3] = "0";
                bases[0, 4] = "0";
                bases[0, 5] = "0";
                bases[0, 6] = "0";
                bases[0, 7] = "0";
                bases[0, 8] = "0";
                bases[0, 9] = "";
                bases[0, 10] = MainForm._mX.ToString();
                bases[0, 11] = MainForm._mY.ToString();
            }
        }

        public void apskaiciuokVisus(int centrineBaze,
double matavimoAukstis)
        {
            suminis = new double[MainForm._Spind * 2 + 1,
MainForm._Spind * 2 + 1];
            MainForm._apskaiciuta = false;
            for (int i = 0; i < suminis.GetLength(0); i++)
            {
                for (int ii = 0; ii < suminis.GetLength(1);
ii++)
                {
                    suminis[i, ii] = 0;
                }
            }
            for (int i = 0; i < bases.GetLength(0); i++)
            {
                if (bases[i, 0] == "@")
                {
                    apskaiciuokAtskirai(i, matavimoAukstis,
centrineBaze);
                }
            }

            apskaiciuokSumini(0);
            MainForm._apskaiciuta = true;
        }

        public void apskaiciuokSumini(int m)
        {
            NumberFormatInfo nfi = new CultureInfo("en-
US", false).NumberFormat;
            nfi.NumberDecimalSeparator = ".";

            Bitmap bmp = new Bitmap(MainForm._Spind * 2,
MainForm._Spind * 2);
            Graphics g = Graphics.FromImage(bmp);
            g.Clear(Color.Transparent);
            Color now;

            System.IO.StreamWriter file = new
System.IO.StreamWriter(failuDirektorija +
"\\suminis.txt");

            double[,] spalvos =
MainForm._spalvos.OrderBy(x => x[3]);

            for (int i = 0; i < suminis.GetLength(0); i++)
            {
                for (int ii = 0; ii < suminis.GetLength(1);
ii++)
                {
                    for (int iz = 0; iz < 5; iz++)

```

```

{
    if (iz != 4 && suminis[i, ii] >= spalvos[iz, 3] && suminis[i, ii] < spalvos[(iz + 1), 3] && spalvos[iz, 3] != 0)
    {
        now = Color.FromArgb((int)(2.55 * MainForm._spalvuPermatomumas), Convert.ToInt32(spalvos[iz, 0]), Convert.ToInt32(spalvos[iz, 1]), Convert.ToInt32(spalvos[iz, 2]));
        g.FillRectangle(new SolidBrush(now), new Rectangle(i, ii, 1, 1));
    }
    else if (iz == 4 && suminis[i, ii] >= spalvos[iz, 3] && spalvos[iz, 3] != 0)
    {
        now = Color.FromArgb((int)(2.55 * MainForm._spalvuPermatomumas), Convert.ToInt32(spalvos[iz, 0]), Convert.ToInt32(spalvos[iz, 1]), Convert.ToInt32(spalvos[iz, 2]));
        g.FillRectangle(new SolidBrush(now), new Rectangle(i, ii, 1, 1));
    }
}
if (m == 0)
{
    file.Write(suminis[i, ii].ToString(nfi) + "
");
}
file.Write("\r\n");
}
file.Close();
g.Flush();
bmp.Save(@failuDirektorija + "\\suminis.png", System.Drawing.Imaging.ImageFormat.Png);
bmp.Dispose();
}

public void apskaiciuokAtskirai(int bazesNr, double matavimoAukstis, int centrineBaze)
{
    NumberFormatInfo nfi = new CultureInfo("en-US", false).NumberFormat;
    nfi.NumberDecimalSeparator = ".";

    string mdPavadinimas = CalculateMD5Hash((bases[basesNr, 1] + bases[basesNr, 2] + bases[basesNr, 3] + bases[basesNr, 4] + bases[basesNr, 5] + bases[basesNr, 6] + bases[basesNr, 7] + bases[basesNr, 8] + bases[basesNr, 9] + bases[basesNr, 10] + bases[basesNr, 11] + matavimoAukstis + MainForm._Kh + MainForm._Kr));
    int startX = (int)Math.Round(Convert.ToDouble(skaiciavimas._bases[centrineBaze, 10])) - MainForm._Spind;
    int startY = (int)Math.Round(Convert.ToDouble(skaiciavimas._bases[centrineBaze, 11])) + MainForm._Spind;

    int nowX = (int)Math.Round(Convert.ToDouble(bases[basesNr, 10], nfi)) - MainForm._Spind;
    int nowY = (int)Math.Round(Convert.ToDouble(bases[basesNr, 11], nfi)) + MainForm._Spind;
    int sumX = nowX - startX;
    int sumY = nowY - startY;
    int tmpX, tmpY;

```

```

    string failoVardas = failuDirektorija + "\\ " + mdPavadinimas + ".txt";

    double[] horizPat = new double[360];
    double[] vertiPat = new double[360];
    if (bases[basesNr, 9].ToString() == "")
    {
        for (int i = 0; i < 359; i++)
        {
            horizPat[i] = 1;
            vertiPat[i] = 1;
        }
    }
    else
    {
        string[] lines = System.IO.File.ReadAllLines(Path.GetDirectoryName(Application.ExecutablePath) + "\\antenas\\ " + bases[basesNr, 9]);
        int hv = 0, hvi = 0;
        foreach (string line in lines)
        {
            string[] lineS = line.Split(' ');
            if (hv == 1 && hvi < 360)
            {
                string[] tmp = lineS[0].Trim().Split('.');
                horizPat[Convert.ToInt32(tmp[0])] = Math.Pow(10, -Convert.ToDouble(lineS[1].Trim(), nfi) / 20);
                hvi++;
            }
            else if (hv == 2 && hvi < 360)
            {
                string[] tmp = lineS[0].Trim().Split('.');
                vertiPat[Convert.ToInt32(tmp[0])] = Math.Pow(10, -Convert.ToDouble(lineS[1].Trim(), nfi) / 20);
                hvi++;
            }
            if (lineS[0].Trim().ToUpper() == "HORIZONTAL")
            {
                hv = 1;
            }
            else if (lineS[0].Trim().ToUpper() == "VERTICAL")
            {
                hv = 2;
                hvi = 0;
            }
        }

        double hMax = horizPat.Max();
        double vMax = vertiPat.Max();

        double z = matavimoAukstis - Convert.ToDouble(bases[basesNr, 2], nfi);
        double Gd = Convert.ToDouble(bases[basesNr, 8], nfi);
        double G = Math.Pow(10, Gd / 10);
        double P = Math.Pow(10, Convert.ToDouble(bases[basesNr, 6], nfi) / 10) / 1000;
        double nFid = Math.Pow(10, -1 * Convert.ToDouble(bases[basesNr, 7], nfi) / 10);
        double nAnt = 1;
        double intensyvumas = Math.Sqrt(30 * P * G * nAnt * nFid);
        double pasukimasFx = 360 - Convert.ToDouble(bases[basesNr, 3], nfi);
        if (pasukimasFx == 360) { pasukimasFx = 0; }
        double palenkimasFy = Convert.ToDouble(bases[basesNr, 4], nfi) + 90;
        for (int y = MainForm._Spind - sumY; y >= -MainForm._Spind - sumY; y--)
        {

```

```

    for (int x = -MainForm._Spind - sumX; x <=
MainForm._Spind - sumX; x++)
    {
        tmpX = sumX + x + MainForm._Spind;
        tmpY = MainForm._Spind*2 - (sumY + y +
MainForm._Spind);
        if ((tmpX >= 0 && tmpX <= MainForm._Spind*2)
&& (tmpY >= 0 && tmpY <= MainForm._Spind*2))
        {
            int absX = x > 0 ? x : -x;
            int absY = y > 0 ? y : -y;
            double absZ = z > 0 ? z : -z;
            double pX = absX * absX;
            double pY = absY * absY;
            double pZ = absZ * absZ;
            double atstumas = (pX + pY);
            double d = Math.Sqrt(atstumas + pZ);

            int Fx = (int)Math.Round(kampas(new double[]
{ x, y }) + pasukimasFx);
            if (Fx > 359) { Fx -= 360; }
            int Fy = (int)Math.Round(kampas(new double[]
{ Math.Sqrt(atstumas), z }) - palenkimasFy);
            if (Fy < 0 ) { Fy += 360; }
            double E = MainForm._Kr * MainForm._Kh *
horizPat[Fx] * vertiPat[Fy] * (intensyvumas /
d);
            double S = Math.Round((1 / 3.77) * E * E,
3);
            suminis[tmpX, tmpY] += S;
        }
    }

    public void eksportuokPolinej()
    {
        int startX =
(int)Math.Round(Convert.ToDouble(skaiciavimas
._bases[MainForm._centrineBaze, 10]));// -
300;
        int startY =
(int)Math.Round(Convert.ToDouble(skaiciavimas
._bases[MainForm._centrineBaze, 11]));//
+300;
    }

    public double[] polar2cartesian(int alfa,
int d)
    {
        double x, y; double tikrasX=0, tikrasY=0;
        if(alfa < 90)
        {
            alfa = 90 - alfa;
            x = Math.Cos(alfa * Math.PI / 180.0) * d;
            y = Math.Sqrt(Math.Pow(d, 2) - Math.Pow(x,
2));

            tikrasX = x;
            tikrasY = y * -1;
        }
        else if(alfa < 180)
        {
            alfa = 180-alfa;
            x = Math.Cos(alfa * Math.PI / 180.0) * d;
            y = Math.Sqrt(Math.Pow(d, 2) - Math.Pow(x,
2));

            tikrasX = y;
            tikrasY = x;
        }
        else if(alfa < 270)
        {
            alfa = 270 - alfa;
            x = Math.Cos(alfa * Math.PI / 180.0) * d;
            y = Math.Sqrt(Math.Pow(d, 2) - Math.Pow(x,
2));

            tikrasX = x * -1;
            tikrasY = y;
        }
        else if(alfa < 360)
        {
            alfa = 360 - alfa;
            x = Math.Cos(alfa * Math.PI / 180.0) * d;
            y = Math.Sqrt(Math.Pow(d, 2) - Math.Pow(x,
2));

            tikrasX = y * -1;
            tikrasY = x * -1;
        }
        return new double[] { Math.Round(tikrasX, 3),
Math.Round(tikrasY, 3) };
    }

    public double kampas(double[] k)
    {
        double theta = Math.Atan2(k[0], k[1]);
        if (theta < 0)
            theta += 2 * 3.14159265359;
        return theta * 180 / 3.14159265359;
    }

    public string CalculateMD5Hash(string input)
    {
        MD5 md5 =
System.Security.Cryptography.MD5.Create();
        byte[] inputBytes =
System.Text.Encoding.ASCII.GetBytes(input);
        byte[] hash = md5.ComputeHash(inputBytes);
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < hash.Length; i++)
        {
            sb.Append(hash[i].ToString("x2"));
        }
        return sb.ToString();
    }

    public string[,] ReDimension(string[, ]
OldArray, int arr1stDimLength, int
arr2ndDimLength)
    {
        string[, ] NewArray = new
string[arr1stDimLength, arr2ndDimLength];
        const int FirstDimension = 0;
        const int SecondDimension = 1;
        int xMax = 0;
        int yMax = 0;
        if (OldArray.GetUpperBound(FirstDimension) <
(arr1stDimLength - 1))
            xMax = OldArray.GetUpperBound(FirstDimension)
+ 1;
        else
            xMax = arr1stDimLength;
        if (OldArray.GetUpperBound(SecondDimension) <
(arr2ndDimLength - 1))
            yMax = OldArray.GetUpperBound(SecondDimension)
+ 1;
        else
            yMax = arr2ndDimLength;
        for (int x = 0; x < xMax; x++)
        {
            for (int y = 0; y < yMax; y++)
            {
                NewArray[x, y] = OldArray[x, y];
            }
        }
        return NewArray;
    }

    public partial class eksportavimas : Form
    {
        private ListViewItem listviewitem;
        skaiciavimas _skaiciavimas = new
skaiciavimas();
        public eksportavimas()
        {
            InitializeComponent();
        }
    }

```

```

        listView1.View =
System.Windows.Forms.View.Details;
        int k = 0;
        this.listView1.Columns.Add("");
        this.listView1.Columns.Add("");
        for (int i = 0; i <= 355; i = i + 5)
        {
            this.listView1.Columns.Add("");
            this.listView1.Columns[k].Width = 50;
            k++;
        }
        this.listView1.Columns[0].Width = 0;
        this.listView1.Columns[1].Width = 50;
    }
    class ListViewNF :
System.Windows.Forms.ListView
    {
        public ListViewNF()

this.SetStyle(ControlStyles.OptimizedDoubleBu
ffer | ControlStyles.AllPaintingInWmPaint,
true);

this.SetStyle(ControlStyles.EnableNotifyMessa
ge, true);
        }
        protected override void
OnNotifyMessage(Message m)
        {
            if (m.Msg != 0x14)
            {
                base.OnNotifyMessage(m);
            }
        }

        public double interpol(double x, double y)
        {
            double P; double R1, R2;
            if (x % 1 == 0 && y % 1 == 0)
            {
                P = skaiciavimas._suminis[(int)x +
MainForm._Spind, (int)y + MainForm._Spind];
            }
            else
            {
                double likX, likY; double x1, x2, y1, y2,
x1N, x2N, y1N, y2N;
                if (x < 0)
                {
                    likX = Math.Ceiling(x) - x;
                }
                else
                {
                    likX = x - Math.Floor(x);
                }
                if (y < 0)
                {
                    likY = Math.Ceiling(y) - y;
                }
                else
                {
                    likY = y - Math.Floor(y);
                }

                x1 = Math.Floor(x);
                y1 = Math.Floor(y);
                x2 = x1 + 1;
                y2 = y1 + 1;
                x1N = skaiciavimas._suminis[(int)x1 +
MainForm._Spind, (int)y1 + MainForm._Spind];
                x2N = skaiciavimas._suminis[(int)x2 +
MainForm._Spind, (int)y1 + MainForm._Spind];
                y1N = skaiciavimas._suminis[(int)x1 +
MainForm._Spind, (int)y2 + MainForm._Spind];
                y2N = skaiciavimas._suminis[(int)x2 +
MainForm._Spind, (int)y2 + MainForm._Spind];
                R1 = ((x2 - x) / (x2 - x1)) * x1N + ((x - x1)
/ (x2 - x1)) * x2N;
                R2 = ((x2 - x) / (x2 - x1)) * y1N + ((x - x1)
/ (x2 - x1)) * y2N;
                P = ((y2 - y) / (y2 - y1)) * R1 + ((y - y1) /
(y2 - y1)) * R2;
            }
            return P;
        }
        int[] d;
        public void rodyk()
        {
            if (MainForm._Spind >= 600)
            {
                d = new int[] { 1, 2, 4, 6, 8, 10, 15, 20, 25,
30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 110,
120, 130, 140, 150, 160, 180, 200, 220, 240,
260, 280, 300, 350, 400, 450, 500, 500, 600 };
            }
            else {
                d = new int[] { 1, 2, 4, 6, 8, 10, 15, 20, 25,
30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 110,
120, 130, 140, 150, 160, 180, 200, 220, 240,
260, 280, 300};
            }
            listView1.Items.Clear();

            skaiciavimas._suminis[0 + MainForm._Spind, 0 +
MainForm._Spind] = skaiciavimas._suminis[1 +
MainForm._Spind, 1 + MainForm._Spind];
            listviewitem = new ListViewItem("");
            listviewitem.SubItems.Add("");
            for (int i = 0; i <= 355; i = i + 5)
            {
                listviewitem.UseItemStyleForSubItems = false;
                listviewitem.SubItems.Add(new
ListViewItem.ListViewSubItem(listviewitem,
i.ToString() + "°", Color.Black,
System.Drawing.SystemColors.Control,
listviewitem.Font));
            }
            this.listView1.Items.Add(listviewitem);

            for (int id = 0; id < d.Length; id++)
            {
                listviewitem = new ListViewItem("");
                listviewitem.UseItemStyleForSubItems = false;
                listviewitem.SubItems.Add(new
ListViewItem.ListViewSubItem(listviewitem,
d[id].ToString() + " m", Color.Black,
System.Drawing.SystemColors.Control,
listviewitem.Font));

                for (int i = 0; i <= 355; i = i + 5)
                {
                    double[] tmp =
_skaiciavimas.polar2cartesian(i, d[id]);
                    double temp = interpol(tmp[0], tmp[1]);

                    if (temp >= 10)
                    {
                        listviewitem.SubItems.Add(new
ListViewItem.ListViewSubItem(listviewitem,
temp.ToString(), Color.White, Color.Red,
listviewitem.Font));
                    }
                    else
                    {
                        listviewitem.SubItems.Add(temp.ToString());
                    }
                }
                this.listView1.Items.Add(listviewitem);
            }
        }
    }

```

```

    }
    private void button1_Click(object sender,
EventArgs e)
    {
        string dt = DateTime.Now.Ticks.ToString();
        System.IO.StreamWriter file = new
        System.IO.StreamWriter(MainForm._direktorija
+ "\\skaiciavimai\\" + dt + ".csv");
        file.Write(";");
        for (int i = 0; i <= 355; i = i + 5)
        {
            file.Write(i.ToString() + ";");
        }
        file.Write("\r\n");
        for (int id = 0; id < d.Length; id++)
        {
            file.Write(d[id] + ";");
            for (int i = 0; i <= 355; i = i + 5)
            {
                double[] tmp =
                _skaiciavimas.polar2cartesian(i, d[id]);
                file.Write(Math.Round(interpol(tmp[0],
                tmp[1]),10).ToString() + ";");
            }
            file.Write("\r\n");
        }
        file.Close();

        System.Diagnostics.Process proc = new
        System.Diagnostics.Process();
        proc.EnableRaisingEvents = false;
        proc.StartInfo.FileName =
        MainForm._direktorija + "\\skaiciavimai\\" +
        dt + ".csv";
        proc.Start();
    }
    private void button2_Click(object sender,
EventArgs e)
    {
        this.Close();
    }
}

public partial class centravimas : Form
{
    public centravimas()
    {
        InitializeComponent();
    }
    public void rodyk()
    {
        NumberFormatInfo nfi = new CultureInfo("en-
        US", false).NumberFormat;
        nfi.NumberDecimalSeparator = ".";
        if (MainForm._Wgs == 0)
        {
            labell.Text = "X, Y (metrais)";
            textBox1.Text = MainForm._mX + ", " +
            MainForm._mY;
        }
        else if (MainForm._Wgs == 1)
        {
            labell.Text = "";
        }
        else if (MainForm._Wgs == 2)
        {
            labell.Text = "Ilguma, platuma (lapsniais)";

            double[] tmp = wgsllks.lks2wgs(new double[] {
            MainForm._mX, MainForm._mY });
            textBox1.Text = Math.Round(tmp[0],
            6).ToString(nfi) + ", " + Math.Round(tmp[1],
            6).ToString(nfi);
        }

        this.comboBox1.Items.Clear();
        this.comboBox1.Items.Add("");

        for(int i=0;
        i<skaiciavimas._bazes.GetLength(0); i++)
        {
            string tt="";
            if (MainForm._Wgs == 0)
            {
                tt = Convert.ToDouble(skaiciavimas._bazes[i,
                10], nfi) +", "+
                Convert.ToDouble(skaiciavimas._bazes[i, 11],
                nfi);
            }
            else if (MainForm._Wgs == 2)
            {
                double[] t = wgsllks.lks2wgs(new double[] {
                Convert.ToDouble(skaiciavimas._bazes[i, 10],
                nfi), Convert.ToDouble(skaiciavimas._bazes[i,
                11], nfi) });
                tt = Math.Round(t[0], 6).ToString(nfi) + ", "
                + Math.Round(t[1], 6).ToString(nfi);
            }

            this.comboBox1.Items.Add(skaiciavimas._bazes[i,
            1] + ", " + skaiciavimas._bazes[i, 3] + "°, " +
            tt);
        }
    }

    private void button1_Click(object sender,
EventArgs e)
    {
        NumberFormatInfo nfi = new CultureInfo("en-
        US", false).NumberFormat;
        nfi.NumberDecimalSeparator = ".";

        if (MainForm._Wgs == 0)
        {
            string[] tmp; double[] lks = new double[2];
            tmp = textBox1.Text.Split(',');
            if (tmp.Length == 2)
            {
                lks = new double[] { Convert.ToDouble(tmp[0],
                nfi), Convert.ToDouble(tmp[1], nfi) };
            }
            else if (tmp.Length == 3)
            {
                tmp = textBox1.Text.Split(' ');
                if (tmp.Length == 2)
                {
                    lks = new double[] {
                    Convert.ToDouble(tmp[0].ToString(nfi)),
                    Convert.ToDouble(tmp[1].ToString(nfi)) };
                }
            }
            else
            {
                tmp = textBox1.Text.Split(' ');
                if (tmp.Length == 2)
                {
                    lks = new double[] { Convert.ToDouble(tmp[0],
                    nfi), Convert.ToDouble(tmp[1], nfi) };
                }
            }
            MainForm.centruok(lks);
        }
        else if (MainForm._Wgs == 1)
        {
        }
        else if (MainForm._Wgs == 2)
        {
            string[] tmp; double[] lks = new double[2];

            tmp = textBox1.Text.Split(',');
            if (tmp.Length == 2)
            {
                lks = new double[] { Convert.ToDouble(tmp[0],
                nfi), Convert.ToDouble(tmp[1], nfi) };
            }

```

```

    }
    else if (tmp.Length == 3)
    {
        tmp = textBox1.Text.Split(' ');
        if (tmp.Length == 2)
        {
            lks = new double[] {
            Convert.ToDouble(tmp[0].ToString(nfi)),
            Convert.ToDouble(tmp[1].ToString(nfi)) };
        }
    }
    else if (tmp.Length == 1)
    {
        tmp = textBox1.Text.Split(' ');
        if (tmp.Length == 2)
        {
            lks = new double[] {
            Convert.ToDouble(tmp[0], nfi),
            Convert.ToDouble(tmp[1], nfi) };
        }
    }
    else
    {
        return;
    }
    MainForm.centruok(wgslks.wgs2lks(lks));
}

if (this.comboBox1.SelectedIndex > 0)
{
    MainForm._mZ = 11;
    MainForm.centruok(new double[] {
    Convert.ToDouble(skaiciavimas._bazes[this.com
   boBox1.SelectedIndex - 1, 10], nfi),
    Convert.ToDouble(skaiciavimas._bazes[this.com
   boBox1.SelectedIndex - 1, 11], nfi) });
}
MainForm.uzdaryk();
this.Close();
}
private void Form_KeyDown(object sender,
KeyEventArgs e)
{
    if (e.KeyCode == Keys.Escape)
    {
        this.Close();
    }
}

public partial class simuliavimas : Form
{
    NumberFormatInfo nfi;
    public simuliavimas()
    {
        InitializeComponent();
        nfi = new CultureInfo("en-US",
        false).NumberFormat;
        nfi.NumberDecimalSeparator = ".";
    }
    skaiciavimas _skaiciavimas = new
    skaiciavimas();
    static double[,] laikinas;
    static double[,] senas;
    static string[,] senos = new string[2, 11];
    public void rodyk()
    {
        this.comboBox1.Items.Clear();

        laikinas = new double[MainForm._Spind * 2 +
        1, MainForm._Spind*2+1];
        senas = new double[MainForm._Spind * 2 + 1,
        MainForm._Spind*2+1];

        for (int i = 0; i <
        skaiciavimas._bazes.GetLength(0); i++)
        {

```

```

        this.comboBox1.Items.Add(skaiciavimas._bazes[i,
        1] + ", " + skaiciavimas._bazes[i, 3] + "°");
        }
        uzpildyk(0);
        this.comboBox1.SelectedIndex = 0;

        senos = _skaiciavimas.ReDimension(senos,
        skaiciavimas._bazes.GetLength(0), 12);

        for (int i = 0; i <
        skaiciavimas._suminis.GetLength(0); i++)
        {
            for (int ii = 0; ii <
            skaiciavimas._suminis.GetLength(1); ii++)
            {
                senas[i, ii] = skaiciavimas._suminis[i, ii];
            }
        }
        for (int i = 0; i <
        skaiciavimas._bazes.GetLength(0); i++)
        {
            senos[i,0] = skaiciavimas._bazes[i, 0];
            senos[i,1] = skaiciavimas._bazes[i, 1];
            senos[i,2] = skaiciavimas._bazes[i, 2];
            senos[i,3] = skaiciavimas._bazes[i, 3];
            senos[i,4] = skaiciavimas._bazes[i, 4];
            senos[i,5] = skaiciavimas._bazes[i, 5];
            senos[i,6] = skaiciavimas._bazes[i, 6];
            senos[i,7] = skaiciavimas._bazes[i, 7];
            senos[i,8] = skaiciavimas._bazes[i, 8];
            senos[i,9] = skaiciavimas._bazes[i, 9];
            senos[i,10] = skaiciavimas._bazes[i, 10];
            senos[i,11] = skaiciavimas._bazes[i, 11];
        }
    }

    public void apskaiciuokAtskirai(int bazesNr,
    double matavimoAukstis, int centrineBaze, int
    m)
    {
        int startX =
        (int)Math.Round(Convert.ToDouble(skaiciavimas._
        bazes[centrineBaze, 10])) - MainForm._Spind;
        int startY =
        (int)Math.Round(Convert.ToDouble(skaiciavimas._
        bazes[centrineBaze, 11])) + MainForm._Spind;

        int nowX =
        (int)Math.Round(Convert.ToDouble(skaiciavimas._
        bazes[basesNr, 10], nfi)) - MainForm._Spind;
        int nowY =
        (int)Math.Round(Convert.ToDouble(skaiciavimas._
        bazes[basesNr, 11], nfi)) + MainForm._Spind;
        int sumX = nowX - startX;
        int sumY = nowY - startY;
        int tmpX, tmpY;

        double[] horizPat = new double[360];
        double[] vertiPat = new double[360];
        if (skaiciavimas._bazes[basesNr, 9].ToString()
        == "")
        {
            for (int i = 0; i < 359; i++)
            {
                horizPat[i] = 1;
                vertiPat[i] = 1;
            }
        }
        else
        {
            string[] lines =
            System.IO.File.ReadAllLines(Path.GetDirectoryNa
            me(Application.ExecutablePath) + "\\antenas\\"
            + skaiciavimas._bazes[basesNr, 9]);
            int hv = 0, hvi = 0;
            foreach (string line in lines)

```

```

{
    string[] lineS = line.Split(' ');
    if (hv == 1 && hvi < 360)
    {
        string[] tmp = lineS[0].Trim().Split('.');
        horizPat[Convert.ToInt32(tmp[0])] =
        Math.Pow(10, -
        Convert.ToDouble(lineS[1].Trim(), nfi) / 20);
        hvi++;
    }
    else if (hv == 2 && hvi < 360)
    {
        string[] tmp = lineS[0].Trim().Split('.');
        vertiPat[Convert.ToInt32(tmp[0])] =
        Math.Pow(10, -
        Convert.ToDouble(lineS[1].Trim(), nfi) / 20);
        hvi++;
    }
    if (lineS[0].Trim().ToUpper() ==
    "HORIZONTAL")
    {
        hv = 1;
    }
    else if (lineS[0].Trim().ToUpper() ==
    "VERTICAL")
    {
        hv = 2;
        hvi = 0;
    }
    }

    double hMax = horizPat.Max();
    double vMax = vertiPat.Max();

    double z = matavimoAukstis -
    Convert.ToDouble(skaiciavimas._bazes[bazesNr,
    2], nfi);
    double Gd =
    Convert.ToDouble(skaiciavimas._bazes[bazesNr,
    8], nfi);
    double G = Math.Pow(10, Gd / 10);
    double P = Math.Pow(10,
    Convert.ToDouble(skaiciavimas._bazes[bazesNr,
    6], nfi) / 10) / 1000;
    double nFid = Math.Pow(10, -1 *
    Convert.ToDouble(skaiciavimas._bazes[bazesNr,
    7], nfi) / 10);
    double nAnt = 1;
    double intensyvumas = Math.Sqrt(30 * P * G *
    nAnt * nFid);
    double pasukimasFx = 360 -
    Convert.ToDouble(skaiciavimas._bazes[bazesNr,
    3], nfi);
    if (pasukimasFx == 360) { pasukimasFx = 0; }
    double palenkimasFy =
    Convert.ToDouble(skaiciavimas._bazes[bazesNr,
    4], nfi) + 90;

    for (int y = MainForm._Spind - sumY; y >= -
    MainForm._Spind - sumY; y--)
    {
        for (int x = -MainForm._Spind - sumX; x <=
        MainForm._Spind - sumX; x++)
        {
            tmpX = sumX + x + MainForm._Spind;
            tmpY = MainForm._Spind*2 - (sumY + y +
            MainForm._Spind);
            if ((tmpX >= 0 && tmpX <= MainForm._Spind *
            2) && (tmpY >= 0 && tmpY <=
            MainForm._Spind*2))
            {
                int absX = x > 0 ? x : -x;
                int absY = y > 0 ? y : -y;
                double absZ = z > 0 ? z : -z;
                double pX = absX * absX;
                double pY = absY * absY;

```

```

                double pZ = absZ * absZ;
                double atstumas = (pX + pY);
                double d = Math.Sqrt(atstumas + pZ);

                int Fx =
                (int)Math.Round(_skaiciavimas.kampas(new
                double[] { x, y }) + pasukimasFx);
                if (Fx > 359) { Fx -= 360; }
                int Fy =
                (int)Math.Round(_skaiciavimas.kampas(new
                double[] { Math.Sqrt(atstumas), z }) -
                palenkimasFy);
                if (Fy < 0) { Fy += 360; }
                double E = MainForm._Kr * MainForm._Kh *
                vertiPat[Fy] * horizPat[Fx] * (intensyvumas /
                d);
                double S = Math.Round((1 / 3.77) * E * E, 3);
                if (m == 0)
                {
                    laikinas[tmpX, tmpY] =
                    skaiciavimas._suminis[tmpX, tmpY] - S;
                }
                else
                {
                    skaiciavimas._suminis[tmpX, tmpY] =
                    laikinas[tmpX, tmpY] + S;
                }
            }
        }
        void uzpildyk(int i)
        {
            textBox5.Text = skaiciavimas._bazes[i,
            2].ToString(nfi);
            textBox6.Text = skaiciavimas._bazes[i,
            3].ToString(nfi);
            textBox7.Text = skaiciavimas._bazes[i,
            4].ToString(nfi);
            textBox1.Text = skaiciavimas._bazes[i,
            6].ToString(nfi);
        }
        private void
        comboBox1_SelectedIndexChanged(object sender,
        EventArgs e)
        {
            int i = this.comboBox1.SelectedIndex;
            if (i >= 0)
            {
                uzpildyk(i);
                apskaiciuokAtskirai(i,
                MainForm._matavimoAukstis,
                MainForm._centrineBaze, 0);
            }
        }
        void simuliuok()
        {
            int i = this.comboBox1.SelectedIndex;
            uzpildyk(i);
            apskaiciuokAtskirai(i,
            MainForm._matavimoAukstis,
            MainForm._centrineBaze, 1);
            _skaiciavimas.apskaiciuokSumini(1);
            MainForm.piesk();
        }
        private void button3_Click(object sender,
        EventArgs e)
        {
            int kur = this.comboBox1.SelectedIndex;
            double nev =
            Convert.ToDouble(skaiciavimas._bazes[kur, 6],
            nfi);
            if (nev > 0) {
                skaiciavimas._bazes[kur, 6] = (nev -
                1).ToString(nfi);
                simuliuok();
            }
        }

```

```

    }
    private void button2_Click(object sender,
EventArgs e)
    {
        int kur = this.comboBox1.SelectedIndex;
        skaiciavimas._bazes[kur, 6] =
(Convert.ToDouble(skaiciavimas._bazes[kur,
6], nfi) + 1).ToString(nfi);
        simuliukok();
    }
    private void button4_Click(object sender,
EventArgs e)
    {
        int kur = this.comboBox1.SelectedIndex;
        double nev =
Convert.ToDouble(skaiciavimas._bazes[kur, 2],
nfi);
        if (nev > 0)
        {
            skaiciavimas._bazes[kur, 2] = (nev -
1).ToString(nfi);
            simuliukok();
        }
        private void button1_Click(object sender,
EventArgs e)
        {
            int kur = this.comboBox1.SelectedIndex;
            skaiciavimas._bazes[kur, 2] =
(Convert.ToDouble(skaiciavimas._bazes[kur,
2], nfi) + 1).ToString(nfi);
            simuliukok();
        }
        private void button6_Click(object sender,
EventArgs e)
        {
            int kur = this.comboBox1.SelectedIndex;
            double nev =
Convert.ToDouble(skaiciavimas._bazes[kur, 3],
nfi);
            if (nev <= 0)
            { nev = 360; }
            skaiciavimas._bazes[kur, 3] = (nev -
5).ToString(nfi);
            simuliukok();
        }
        private void button5_Click(object sender,
EventArgs e)
        {
            int kur = this.comboBox1.SelectedIndex;
            double nev =
Convert.ToDouble(skaiciavimas._bazes[kur, 3],
nfi);
            if (nev >= 359)
            { nev = -1; }
            skaiciavimas._bazes[kur, 3] = (nev +
5).ToString(nfi);
            simuliukok();
        }
        private void button8_Click(object sender,
EventArgs e)
        {
            int kur = this.comboBox1.SelectedIndex;
            double nev =
Convert.ToDouble(skaiciavimas._bazes[kur, 4],
nfi);
            if (nev > -44)
            {
                skaiciavimas._bazes[kur, 4] = (nev -
1).ToString(nfi);
                simuliukok();
            }
        }
        private void button7_Click(object sender,
EventArgs e)
        {
            int kur = this.comboBox1.SelectedIndex;

```

```

        double nev =
Convert.ToDouble(skaiciavimas._bazes[kur, 4],
nfi);
        if (nev < 45)
        {
            skaiciavimas._bazes[kur, 4] = (nev +
1).ToString(nfi);
            simuliukok();
        }
    }
    private void Form_KeyDown(object sender,
KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            int i = this.comboBox1.SelectedIndex;
            skaiciavimas._bazes[i, 2] =
textBox5.Text.ToString(nfi);
            skaiciavimas._bazes[i, 3] =
textBox6.Text.ToString(nfi);
            skaiciavimas._bazes[i, 4] =
textBox7.Text.ToString(nfi);
            skaiciavimas._bazes[i, 6] =
textBox1.Text.ToString(nfi);
            simuliukok();
        }
        if (e.KeyCode == Keys.Escape)
        {
            uzdaryk();
        }
    }
    void uzdaryk()
    {
        for (int i = 0; i <
skaiciavimas._suminis.GetLength(0); i++)
        {
            for (int ii = 0; ii <
skaiciavimas._suminis.GetLength(1); ii++)
            {
                skaiciavimas._suminis[i, ii] = senas[i, ii];
            }
        }
        for (int i = 0; i <
skaiciavimas._bazes.GetLength(0); i++)
        {
            skaiciavimas._bazes[i, 0] = senos[i, 0];
            skaiciavimas._bazes[i, 1] = senos[i, 1];
            skaiciavimas._bazes[i, 2] = senos[i, 2];
            skaiciavimas._bazes[i, 3] = senos[i, 3];
            skaiciavimas._bazes[i, 4] = senos[i, 4];
            skaiciavimas._bazes[i, 5] = senos[i, 5];
            skaiciavimas._bazes[i, 6] = senos[i, 6];
            skaiciavimas._bazes[i, 7] = senos[i, 7];
            skaiciavimas._bazes[i, 8] = senos[i, 8];
            skaiciavimas._bazes[i, 9] = senos[i, 9];
            skaiciavimas._bazes[i, 10] = senos[i, 10];
            skaiciavimas._bazes[i, 11] = senos[i, 11];
        }
        _skaiciavimas.apskaiciuokSumini(1);
        MainForm.piesk();
        this.Close();
    }
    private void button9_Click(object sender,
EventArgs e)
    {
        uzdaryk();
    }
    private void button9_Click_1(object sender,
EventArgs e)
    {
        this.Close();
    }
    private void textBox_KeyPress(object sender,
KeyPressEventArgs e)
    {

```

```

        if (!char.IsControl(e.KeyChar) &&
!char.IsDigit(e.KeyChar) && e.KeyChar != '.')
        {
            e.Handled = true;
        }
        if (e.KeyChar == '.' && (sender as
TextBox).Text.IndexOf('.') > -1)
        {
            e.Handled = true;
        }
    }

    public partial class minimizavimas : Form
    {
        skaiciavimas _skaiciavimas = new
skaiciavimas();
        stotys _stotys = new stotys();
        static string[] miniBazes = new string[0];
        static double[,] taskai = new double[4,3];
        static double[,] masyvas = new double[2,
11];
        static double[,] masyvas2 = new double[2,
5];
        static int taskaiNr = 0;
        public static double[,] _taskai
        {
            get { return taskai; }
            set { taskai = value; }
        }
        public static int _taskaiNr
        {
            get { return taskaiNr; }
            set { taskaiNr = value; }
        }
        int listView2pazymetas = -1;
        NumberFormatInfo nfi;
        private ListViewItem listviewitem;

        public minimizavimas()
        {
            InitializeComponent();
            listView1.View =
System.Windows.Forms.View.Details;
            this.listView1.Columns.Add("");
            this.listView1.Columns[0].Width = 116;
            listView2.View =
System.Windows.Forms.View.Details;
            this.listView2.Columns.Add("");
            this.listView2.Columns[0].Width = 116;
            listView3.View =
System.Windows.Forms.View.Details;
            this.listView3.Columns.Add("");
            this.listView3.Columns[0].Width = 116;
            listView4.View =
System.Windows.Forms.View.Details;
            this.listView4.Columns.Add("Nr");
            this.listView4.Columns.Add("X");
            this.listView4.Columns.Add("Y");
            this.listView4.Columns.Add("Aukštis, m");
            this.listView4.Columns.Add("Kryptis, °");
            this.listView4.Columns.Add("Palenkimas, °");
            this.listView4.Columns.Add("Galia, dBm");
            this.listView4.Columns.Add("Vidurkis
μW/cm²");
            this.listView4.Columns.Add("Nr. 1 μW/cm²");
            this.listView4.Columns.Add("Nr. 2 μW/cm²");
            this.listView4.Columns.Add("Nr. 3 μW/cm²");
            this.listView4.Columns.Add("Hata");
            this.listView4.Columns[0].Width = 50;
            nfi = new CultureInfo("en-US",
false).NumberFormat;
            nfi.NumberDecimalSeparator = ".";
        }
        public void rodyk()
        {
            listView1.Items.Clear();
            for (int i = 0; i <
skaiciavimas._bases.GetLength(0); i++)
            {
                ListViewItem listviewitem = new
ListViewItem(skaiciavimas._bases[i, 1] + ", " +
skaiciavimas._bases[i, 3] + "°");
                this.listView1.Items.Add(listviewitem);
            }

            if (MainForm._Wgs == 0)
            {
                if (taskai[1, 0] > 0) textBox11.Text =
taskai[1, 0].ToString(nfi) + ", " + taskai[1,
1].ToString(nfi);
                if (taskai[2, 0] > 0) textBox13.Text =
taskai[2, 0].ToString(nfi) + ", " + taskai[2,
1].ToString(nfi);
                if (taskai[3, 0] > 0) textBox15.Text =
taskai[3, 0].ToString(nfi) + ", " + taskai[3,
1].ToString(nfi);
            }
            else if (MainForm._Wgs == 2)
            {
                if (taskai[1, 0] > 0)
                {
                    double[] tmp = wgslks.lks2wgs(new double[] {
taskai[1, 0], taskai[1, 1] });
                    textBox11.Text = Math.Round(tmp[0],
6).ToString(nfi) + ", " + Math.Round(tmp[1],
6).ToString(nfi);
                }
                if (taskai[2, 0] > 0)
                {
                    double[] tmp = wgslks.lks2wgs(new double[] {
taskai[2, 0], taskai[2, 1] });
                    textBox13.Text = Math.Round(tmp[0],
6).ToString(nfi) + ", " + Math.Round(tmp[1],
6).ToString(nfi);
                }
                if (taskai[3, 0] > 0)
                {
                    double[] tmp = wgslks.lks2wgs(new double[] {
taskai[3, 0], taskai[3, 1] });
                    textBox15.Text = Math.Round(tmp[0],
6).ToString(nfi) + ", " + Math.Round(tmp[1],
6).ToString(nfi);
                }
            }
            private void listView2_MouseUp(object sender,
MouseEventArgs e)
            {
                for (int i = 0; i < listView2.Items.Count;
i++)
                {
                    if (listView2.Items[i].Selected == true)
                    {
                        listView2pazymetas = i;
                        listView3.Items.Clear();
                        string[] words = miniBazes[i].Split(',');
                        foreach (string word in words)
                        {
                            if (word.Length > 0)
                            {
                                string text =
skaiciavimas._bases[Convert.ToInt32(word), 1] +
", " +
skaiciavimas._bases[Convert.ToInt32(word), 3] +
"°";
                                ListViewItem listviewitem = new
ListViewItem(text);
                                this.listView3.Items.Add(listviewitem);
                            }
                        }
                    }
                }
            }
        }
    }

```

```

    }
    private void button2_Click(object sender,
EventArgs e)
    {
        for (int i = 0; i < listView1.Items.Count;
i++)
        {
            if (listView1.Items[i].Selected == true)
            {
                if (listview2pazymetas >= 0)
                {
                    miniBazes[listview2pazymetas] +=
i.ToString() + ",";
                    listView3.Items.Clear();
                    string[] words =
miniBazes[listview2pazymetas].Split(',');
                    foreach (string word in words)
                    {
                        if (word.Length > 0)
                        {
                            string text =
skaiciavimas._bazes[Convert.ToInt32(word), 1]
+ ", " +
skaiciavimas._bazes[Convert.ToInt32(word), 3]
+ "°";
                            ListViewItem listviewitem = new
ListViewItem(text);
                            this.listView3.Items.Add(listviewitem);
                        }
                    }
                }
                else
                {
                    Array.Resize(ref miniBazes,
miniBazes.GetLength(0) + 1);
                    miniBazes[miniBazes.GetLength(0) - 1] =
i.ToString() + ",";
                    listView2.Items.Clear();
                    for (int k = 0; k < miniBazes.GetLength(0);
k++)
                    {
                        string[] words = miniBazes[k].Split(',');
                        string text =
skaiciavimas._bazes[Convert.ToInt32(words[0])
, 1] + ", " +
skaiciavimas._bazes[Convert.ToInt32(words[0])
, 3] + "°";
                        ListViewItem listviewitem = new
ListViewItem(text);
                        this.listView2.Items.Add(listviewitem);
                    }
                }
            }
        }
        private void button1_Click(object sender,
EventArgs e)
        {
            listview2pazymetas = -1;
        }
        public double[,] ReDimension(double[, ]
OldArray, int arr1stDimLength, int
arr2ndDimLength)
        {
            double[, ] NewArray = new
double[arr1stDimLength, arr2ndDimLength];
            const int FirstDimension = 0;
            const int SecondDimension = 1;
            int xMax = 0;
            int yMax = 0;
            if (OldArray.GetUpperBound(FirstDimension) <
(arr1stDimLength - 1))
                xMax =
OldArray.GetUpperBound(FirstDimension) + 1;
            else
                xMax = arr1stDimLength;

```

```

            if (OldArray.GetUpperBound(SecondDimension) <
(arr2ndDimLength - 1))
                yMax = OldArray.GetUpperBound(SecondDimension)
+ 1;
            else
                yMax = arr2ndDimLength;
            for (int x = 0; x < xMax; x++)
            {
                for (int y = 0; y < yMax; y++)
                {
                    NewArray[x, y] = OldArray[x, y];
                }
            }
            return NewArray;
        }
        public double apskaiciuokAtskirai(int bazesNr,
int oPx, int oPy, int oH, int oD, int oT, int
oS, double matavimoAukstis, int[] kords)
        {
            int nowX =
(int)Math.Round(Convert.ToDouble(skaiciavimas._
bazes[bazesNr, 10], nfi)) + oPx;
            int nowY =
(int)Math.Round(Convert.ToDouble(skaiciavimas._
bazes[bazesNr, 11], nfi)) + oPy;
            int x = nowX - kords[0];
            int y = nowY - kords[1];

            double[] horizPat = new double[360];
            double[] vertiPat = new double[360];
            if (skaiciavimas._bazes[bazesNr, 9].ToString()
== "")
            {
                for (int i = 0; i < 359; i++)
                {
                    horizPat[i] = 1;
                    vertiPat[i] = 1;
                }
            }
            else
            {
                string[] lines =
System.IO.File.ReadAllLines(Path.GetDirectoryNa
me(Application.ExecutablePath) + "\\antenos\\"
+ skaiciavimas._bazes[bazesNr, 9]);
                int hv = 0, hvi = 0;
                foreach (string line in lines)
                {
                    string[] lineS = line.Split(' ');
                    if (hv == 1 && hvi < 360)
                    {
                        string[] tmp = lineS[0].Trim().Split('.');
                        horizPat[Convert.ToInt32(tmp[0])] =
Math.Pow(10, -Convert.ToDouble(lineS[1].Trim(),
nfi) / 20);
                        hvi++;
                    }
                    else if (hv == 2 && hvi < 360)
                    {
                        string[] tmp = lineS[0].Trim().Split('.');
                        vertiPat[Convert.ToInt32(tmp[0])] =
Math.Pow(10, -Convert.ToDouble(lineS[1].Trim(),
nfi) / 20);
                        hvi++;
                    }
                    if (lineS[0].Trim().ToUpper() == "HORIZONTAL")
                    {
                        hv = 1;
                    }
                    else if (lineS[0].Trim().ToUpper() ==
"VERTICAL")
                    {
                        hv = 2;
                        hvi = 0;
                    }
                }
            }

```

```

double hMax = horizPat.Max();
double vMax = vertiPat.Max();

double z = matavimoAukstis -
Convert.ToDouble(skaiciavimas._bazes[bazesNr,
2], nfi) - oH;
double Gd =
Convert.ToDouble(skaiciavimas._bazes[bazesNr,
8], nfi);
double G = Math.Pow(10, Gd / 10);
double P = Math.Pow(10,
(Convert.ToDouble(skaiciavimas._bazes[bazesNr,
6], nfi) + oS) / 10) / 1000;
double nFid = Math.Pow(10, -1 *
Convert.ToDouble(skaiciavimas._bazes[bazesNr,
7], nfi) / 10);
double nAnt = 1;
double intensyvumas = Math.Sqrt(30 * P * G *
nAnt * nFid);
double pasukimasFx = 360 -
Convert.ToDouble(skaiciavimas._bazes[bazesNr,
3], nfi) + oD;
if (pasukimasFx == 360) { pasukimasFx = 0; }
double palenkimasFy =
Convert.ToDouble(skaiciavimas._bazes[bazesNr,
4], nfi) + 90 + oT;
int absX = x > 0 ? x : -x;
int absY = y > 0 ? y : -y;
double absZ = z > 0 ? z : -z;
double pX = absX * absX;
double pY = absY * absY;
double pZ = absZ * absZ;
double atstumas = (pX + pY);
double d = Math.Sqrt(atstumas + pZ);

int Fx =
(int)Math.Round(_skaiciavimas.kampas(new
double[] { absX, absY }) + pasukimasFx);
if (x <= 0 && y > 0) { Fx = 180 - Fx; }
else if (x > 0 && y > 0) { Fx = 180 + Fx; }
else if (x >= 0 && y <= 0) { Fx = 360 - Fx; }
if (Fx > 359) { Fx -= 360; }
else if (Fx < 0) { Fx += 360; }
int Fy =
(int)Math.Round(_skaiciavimas.kampas(new
double[] { Math.Sqrt(atstumas), z }) -
palenkimasFy);
if (Fy < 0) { Fy += 360; }

double E = MainForm.Kr * MainForm.Kh *
vertiPat[Fy] * horizPat[Fx] * (intensyvumas /
d);

return Math.Round((1 / 3.77) * E * E, 8);
}

public double apskaiciuokAtskiraiHata(int
bazesNr, int oPx, int oPy, int oH, int oD,
int oT, int oS, double matavimoAukstis,
double atstumas)
{
double[] horizPat = new double[360];
double[] vertiPat = new double[360];
double f=1000;
if (skaiciavimas._bazes[bazesNr,
9].ToString() == "")
{
for (int i = 0; i < 359; i++)
{
horizPat[i] = 1;
vertiPat[i] = 1;
}
}
else

```

```

{
string[] lines =
System.IO.File.ReadAllLines(Path.GetDirectoryNa
me(Application.ExecutablePath) + "\\antenas\\"
+ skaiciavimas._bazes[bazesNr, 9]);
int hv = 0, hvi = 0;
foreach (string line in lines)
{
string[] lineS = line.Split(' ');
if (hv == 1 && hvi < 360)
{
string[] tmp = lineS[0].Trim().Split('.');
horizPat[Convert.ToInt32(tmp[0])] =
Convert.ToDouble(lineS[1].Trim(), nfi);
hvi++;
}
else if (hv == 2 && hvi < 360)
{
string[] tmp = lineS[0].Trim().Split('.');
vertiPat[Convert.ToInt32(tmp[0])] =
Convert.ToDouble(lineS[1].Trim(), nfi);
hvi++;
}
if (lineS[0].Trim().ToUpper() == "FREQUENCY")
{
f = Convert.ToDouble(lineS[1].Trim(), nfi);
}
else if (lineS[0].Trim().ToUpper() ==
"HORIZONTAL")
{
hv = 1;
}
else if (lineS[0].Trim().ToUpper() ==
"VERTICAL")
{
hv = 2;
hvi = 0;
}
}
double ht =
Math.Log(Convert.ToDouble(skaiciavimas._bazes[b
azesNr, 2], nfi) + oH, 10);
double d = Math.Log(atstumas, 10);
double hm = matavimoAukstis;
f = Math.Log(f, 10);

double G = Math.Pow(10,
Convert.ToDouble(skaiciavimas._bazes[bazesNr,
8], nfi) / 10);
double P = Math.Pow(10,
(Convert.ToDouble(skaiciavimas._bazes[bazesNr,
6], nfi) + oS) / 10);
double nFid = Math.Pow(10, -1 *
Convert.ToDouble(skaiciavimas._bazes[bazesNr,
7], nfi) / 10);
double nAnt = 0.9;
double stiprumas = 10 * Math.Log(P * G * nFid
* nAnt, 10);
int Fx = 360 + oD;
if (Fx > 359) { Fx -= 360; }

double z = matavimoAukstis -
Convert.ToDouble(skaiciavimas._bazes[bazesNr,
2], nfi) + oH;
double palenkimasFy =
Convert.ToDouble(skaiciavimas._bazes[bazesNr,
4], nfi) + oT + 90;
int Fy =
(int)Math.Round(_skaiciavimas.kampas(new
double[] { atstumas*1000, z }) - palenkimasFy);
if (Fy < 0) { Fy += 360; }
stiprumas = stiprumas - horizPat[Fx] -
vertiPat[Fy];

```

```

    double slopinimas = 69.55 + 26.16 * f -
13.82 * ht - ((1.1 * f - 0.7) * hm - (1.56 *
f - 0.8)) + (44.9 - 6.55 * ht) * d;
    return stiprumas-slopinimas;
}

    string failuDirektorija =
MainForm._direktorija + "\\skaiciavimai";

    private void button4_Click(object sender,
EventArgs e)
    {
        System.IO.StreamWriter file = new
System.IO.StreamWriter(failuDirektorija +
"\\minimiz.txt");

        double[] taskaiTmp = new double[2];
        taskaiTmp =
        _stotys.string2kords(textBox11.Text);
        taskai[1, 0] = taskaiTmp[0];
        taskai[1, 1] = taskaiTmp[1];
        if(textBox12.Text.Length > 0) taskai[1, 2] =
Convert.ToDouble(textBox12.Text);
        taskaiTmp =
        _stotys.string2kords(textBox13.Text);
        taskai[2, 0] = taskaiTmp[0];
        taskai[2, 1] = taskaiTmp[1];
        if (textBox14.Text.Length > 0) taskai[2, 2]
= Convert.ToDouble(textBox14.Text);
        taskaiTmp =
        _stotys.string2kords(textBox15.Text);
        taskai[3, 0] = taskaiTmp[0];
        taskai[3, 1] = taskaiTmp[1];
        if (textBox16.Text.Length > 0) taskai[3, 2]
= Convert.ToDouble(textBox16.Text);

        int oPxT = Convert.ToInt32(textBox1.Text) /
2;
        int oPyT = Convert.ToInt32(textBox2.Text) /
2;
        int oHxT = Convert.ToInt32(textBox3.Text);
        int oHyT = Convert.ToInt32(textBox4.Text);
        int oDxT = Convert.ToInt32(textBox5.Text);
        int oDyT = Convert.ToInt32(textBox6.Text);
        int oTxT = Convert.ToInt32(textBox7.Text);
        int oTyT = Convert.ToInt32(textBox8.Text);
        int oSxT = Convert.ToInt32(textBox9.Text);
        int oSyT = Convert.ToInt32(textBox10.Text);
        int m = 0;
        apskaiciuokAtskirai(0, 0, 0, 0, 0, 0, 0, 0,
taskai[1, 2], new int[] {
Convert.ToInt32(taskai[1, 0]),
Convert.ToInt32(taskai[1, 1]) });

        int BSkiekis = 0;
        for (int k = 0; k < miniBazes.GetLength(0);
k++)
        {
            string[] words = miniBazes[k].Split(',');
            BSkiekis += words.GetLength(0) - 1;
        }
        masyvas2 = ReDimension(masyvas2,
skaiciavimas._bazes.GetLength(0) + 1, 5);

        for (int k = 0; k < miniBazes.GetLength(0);
k++)
        {
            string[] words = miniBazes[k].Split(',');
            for (int BS = 0; BS < words.GetLength(0) -
1; BS++)
            {
                int i = Convert.ToInt32(words[BS]);
                if (taskai[1, 0] > 0) masyvas2[i, 0] =
apskaiciuokAtskirai(i, 0, 0, 0, 0, 0, 0, 0,
taskai[1, 2], new int[] {
Convert.ToInt32(taskai[1, 0]),
Convert.ToInt32(taskai[1, 1]) });
                if (taskai[2, 0] > 0) masyvas2[i, 1] =
apskaiciuokAtskirai(i, 0, 0, 0, 0, 0, 0, 0,
taskai[2, 2], new int[] {
Convert.ToInt32(taskai[2, 0]),
Convert.ToInt32(taskai[2, 1]) });
                if (taskai[3, 0] > 0) masyvas2[i, 2] =
apskaiciuokAtskirai(i, 0, 0, 0, 0, 0, 0, 0,
taskai[3, 2], new int[] {
Convert.ToInt32(taskai[3, 0]),
Convert.ToInt32(taskai[3, 1]) });
                masyvas[m, 0] = i;
                masyvas[m, 1] =
Convert.ToDouble(skaiciavimas._bazes[i, 2]) +
oH;
                masyvas[m, 2] =
Convert.ToDouble(skaiciavimas._bazes[i, 3]) +
oD;
                masyvas[m, 3] =
Convert.ToDouble(skaiciavimas._bazes[i, 4]) +
oT;
                masyvas[m, 4] =
Convert.ToDouble(skaiciavimas._bazes[i, 6]) +
oS;
                masyvas[m, 5] =
Convert.ToDouble(skaiciavimas._bazes[i, 10]) +
oPx;
            }
        }
    }
}

```

```

    masyvas[m, 6] =
    Convert.ToDouble(skaiciavimas._bazes[i, 11])
    + oPy;
    masyvas[m, 10] = apskaiciuokAtskiraiHata(i,
    0, 0, oH, oD, oT, oS, 1.5, 5);
    file.Write(i.ToString() + " " + masyvas[m,
    1].ToString() + " " + masyvas[m,
    2].ToString() + " " + masyvas[m,
    3].ToString() + " " + masyvas[m,
    4].ToString() + " " + masyvas[m,
    7].ToString() + " " + masyvas[m,
    8].ToString() + " " + masyvas[m,
    9].ToString() + " " + masyvas[m, 10]
    + "\r\n");
    m++;
    }
    }
    }
    }
    }
    }
    file.Write("++\r\n");
    file.Close();
    parodyk();
    }
    public void parodyk()
    {
        int tskKiekis = taskai[1, 0] > 0 ? 1 : 0;
        tskKiekis += taskai[2, 0] > 0 ? 1 : 0;
        tskKiekis += taskai[3, 0] > 0 ? 1 : 0;
        listView4.Items.Clear();
        Color greyColor = Color.FromArgb(220, 220,
        220);
        int spalvotaEile = 1;
        for (int k = 0; k < miniBazes.GetLength(0);
        k++)
        {
            string[] words = miniBazes[k].Split(',');
            for (int l = 0; l < words.GetLength(0) - 1;
            l++)
            {
                int kuris = 0;
                double yra = 999;
                double yra1 = 999, yra2 = 999, yra3 = 999;
                // int BS = Convert.ToInt32(masyvas[kuris,
                0]);
                int BS = Convert.ToInt32(words[l]);
                for (int n = 0; n < masyvas.GetLength(0);
                n++)
                {
                    int t = 0, d = 0;
                    double Ssuma = (masyvas[n, 7] + masyvas[n,
                    8] + masyvas[n, 9]) / tskKiekis;
                    // if (Ssuma < yra && masyvas[n, 0] == BS)
                    if (masyvas[n, 7] < yra1 && masyvas[n, 8] <
                    yra2 && masyvas[n, 9] < yra3 && masyvas[n, 0]
                    == BS)
                    {
                        t += (!checkBox2.Checked &&
                        Convert.ToDouble(skaiciavimas._bazes[BS, 2])
                        == masyvas[n, 1]) ? 1 : 0;
                        t += (!checkBox3.Checked &&
                        Convert.ToDouble(skaiciavimas._bazes[BS, 3])
                        == masyvas[n, 2]) ? 2 : 0;
                        t += (!checkBox4.Checked &&
                        Convert.ToDouble(skaiciavimas._bazes[BS, 4])
                        == masyvas[n, 3]) ? 4 : 0;
                        t += (!checkBox5.Checked &&
                        Convert.ToDouble(skaiciavimas._bazes[BS, 6])
                        == masyvas[n, 4]) ? 8 : 0;
                        t += (!checkBox1.Checked &&
                        Convert.ToDouble(skaiciavimas._bazes[BS, 10])
                        == masyvas[n, 5] &&
                        Convert.ToDouble(skaiciavimas._bazes[BS, 11])
                        == masyvas[n, 6]) ? 16 : 0;

```

```

        d += !checkBox2.Checked ? 1 : 0;
        d += !checkBox3.Checked ? 2 : 0;
        d += !checkBox4.Checked ? 4 : 0;
        d += !checkBox5.Checked ? 8 : 0;
        d += !checkBox1.Checked ? 16 : 0;

        if (t == d)
        {
            yra1 = masyvas[n, 7];
            yra2 = masyvas[n, 8];
            yra3 = masyvas[n, 9];
            yra = Ssuma;
            kuris = n;
        }
        }

        string text = skaiciavimas._bazes[BS, 1] + "
        " + skaiciavimas._bazes[BS, 3] + "°";
        listviewitem = new ListViewItem(text);
        listviewitem.SubItems.Add(masyvas[kuris,
        5].ToString());
        listviewitem.SubItems.Add(masyvas[kuris,
        6].ToString());
        listviewitem.SubItems.Add(masyvas[kuris,
        1].ToString());
        listviewitem.SubItems.Add(masyvas[kuris,
        2].ToString());
        listviewitem.SubItems.Add(masyvas[kuris,
        3].ToString());
        listviewitem.SubItems.Add(masyvas[kuris,
        4].ToString());

        listviewitem.SubItems.Add(Math.Round(yra, 3).ToS
        tring());
        listviewitem.SubItems.Add(masyvas[kuris,
        7].ToString());
        listviewitem.SubItems.Add(masyvas[kuris,
        8].ToString());
        listviewitem.SubItems.Add(masyvas[kuris,
        9].ToString());
        listviewitem.SubItems.Add(masyvas[kuris,
        10].ToString());
        this.listView4.Items.Add(listviewitem);
        listviewitem = new ListViewItem(text);

        listviewitem.SubItems.Add(skaiciavimas._bazes[B
        S, 10]);

        listviewitem.SubItems.Add(skaiciavimas._bazes[B
        S, 11]);

        listviewitem.SubItems.Add(skaiciavimas._bazes[B
        S, 2]);

        listviewitem.SubItems.Add(skaiciavimas._bazes[B
        S, 3]);

        listviewitem.SubItems.Add(skaiciavimas._bazes[B
        S, 4]);

        listviewitem.SubItems.Add(skaiciavimas._bazes[B
        S, 6]);

        listviewitem.SubItems.Add(Math.Round((masyvas2[
        BS, 0] + masyvas2[BS, 1] + masyvas2[BS, 2]) /
        tskKiekis, 3).ToString());
        listviewitem.SubItems.Add(masyvas2[BS,
        0].ToString());
        listviewitem.SubItems.Add(masyvas2[BS,
        1].ToString());
        listviewitem.SubItems.Add(masyvas2[BS,
        2].ToString());
        listviewitem.SubItems.Add(masyvas2[BS,
        3].ToString());
        this.listView4.Items.Add(listviewitem);

```

```

        listView4.Items[spalvotaEile].BackColor =
greyColor;
        spalvotaEile = spalvotaEile + 2;
    }
}
private void textBox11_MouseUp(object
sender, MouseEventArgs e)
{
    taskaiNr = 1;
    if(taskai[1, 0] > 0) MainForm.centruok(new
double[] { taskai[1, 0], taskai[1, 1] });
    this.Close();
}
private void textBox13_MouseUp(object
sender, MouseEventArgs e)
{
    taskaiNr = 2;
    if (taskai[2, 0] > 0) MainForm.centruok(new
double[] { taskai[2, 0], taskai[2, 1] });
    this.Close();
}
private void textBox15_MouseUp(object
sender, MouseEventArgs e)
{
    taskaiNr = 3;

```

```

        if (taskai[3, 0] > 0) MainForm.centruok(new
double[] { taskai[3, 0], taskai[3, 1] });
        this.Close();
    }
    private void textBox_KeyPress(object sender,
KeyPressEventArgs e)
    {
        if (!char.IsControl(e.KeyChar) &&
!char.IsDigit(e.KeyChar) && e.KeyChar != '.')
        {
            e.Handled = true;
        }
        if (e.KeyChar == '.' && (sender as
TextBox).Text.IndexOf('.') > -1)
        {
            e.Handled = true;
        }
    }

    private void checkBox1_CheckedChanged(object
sender, EventArgs e)
    {
        parodyk();
    }
}

```

**D PRIEDAS. PAŽYMA APIE DALYVAVIMĄ XVI-OJE LIETUVOS
JAUNŲJŲ MOKSLININKŲ KONFERENCIJOJE „MOKSLAS – LIETUVOS
ATEITIS“**