



KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS

Mindaugas Vičiulis

**LINIJINIO POSLINKIO ABSOLIUTINIŲ KEITIKLIŲ
ALGORITMŲ, SKIRTŲ PADĖTIES NUSTATYMOI, TYRIMAS**

Baigiamasis magistro projektas

Vadovas

Prof. dr. Žilvinas Nakutis

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS
ELEKTRONIKOS INŽINERIJOS KATEDRA

**LINIJINIO POSLINKIO ABSOLIUTINIŲ KEITIKLIŲ
ALGORITMŲ, SKIRTŲ PADĖTIES NUSTATYMOI, TYRIMAS**

Baigiamasis magistro projektas
Įterptinės sistemos M2146M21

Vadovas

Prof. dr. Žilvinas Nakutis

Recenzentas

Prof. dr. Vytautas Deksnys

Projektą atliko

Mindaugas Vičiulis

KAUNAS, 2016



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Elektros ir elektronikos fakultetas

(Fakultetas)

Mindaugas Vičiulis

(Studento vardas, pavardė)

Įterptinės sistemos M2146M21

(Studijų programos pavadinimas, kodas)

Linijinio poslinkio absoliutinių keitiklių algoritmų, skirtų padėties nustatymui,
tyrimas

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 16 m. 05 24 d.
Kaunas

Patvirtinu, kad mano, **Mindaugo Vičiulio**, baigiamasis projektas tema „LINIJINIO POSLINKIO ABSOLIUTINIŲ KEITIKLIŲ ALGORITMŲ, SKIRTŲ PADĖTIES NUSTATYMO, TYRIMAS“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

(vardą ir pavardę įrašyti ranka)

(parašas)

Vičiulis, M. Linijinio poslinkio absoliutinių keitiklių algoritmų, skirtų padėties nustatymui, tyrimas. Magistro baigiamasis projektas, vadovas Prof. dr. Žilvinas Nakutis; Kauno technologijos universitetas, Elektros ir Elektronikos fakultetas, Elektronikos inžinerijos katedra.

Kaunas, 2016. 47 psl.

SANTRAUKA

Darbe nagrinėjamas absoliutinis padėties nustatymas naudojant pseudo atsitiktinę seką. Tiriama trys padėties nustatyto algoritmai paremti Euklido atstumu, koreliacija ir LFSR (Linear Feedback Shift Register) generatoriais. Atliekant darbą pasiūlytos algoritmų modifikacijos mažinančios klaidų įtaką bei didinančios greitaveiką. Sukuriamas algoritmų modelis pasinaudojant Matlab programiniu paketu. Algoritmai tiriama atsparumui pavienėms ir grupinėms klaidoms, tai pat tiriama algoritmų greitaveiką. Gauti rezultatai lyginami tarpusavyje.

Reikšminiai žodžiai (iki 8 žodžių): LFSR, koreliacija, Euklidas, Pseudo atsitiktine seka, Padėties aptikimas.

Viciulis, Minduagas. Research of algorithms for linear absolute encoders position detection. Masters *degree*. supervisor prof. dr. Zilvinas Nakutis; Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of Electronics.

Kaunas, 2016. 50 psl.

SUMMARY

The paper discusses the use of pseudo random sequence to determine absolute position. Examined the three algorithm which are based on the Euclidean distance, correlation and LFSR (Linear Feedback Shift Register) generators. The work proposes algorithm modifications reducing errors and increasing the speed of execution. There are created model of algorithms, using Matlab software package. The resilience of algorithms is tested individual and group errors, the speed of algorithms is also examined. The results obtained are compared with each other.

Keywords (up to 8 words): LFSR, Correlation, Euclid, a pseudo random sequence, positioning match.

TURINYS

ĮVADAS	7
1. ABSOLIUČIOJO ATSTUMO MATAVIMŲ METODŲ APŽVALGA	8
1.1. Enkoderių rūšys	8
1.2. Inkrementiniai enkoderiai	8
1.3. Absolūtieji enkoderiai	9
1.4. Nuoseklaus kodavimo būdai	10
2. LINIJINIO ABSOLIUTINIO POSLINKIO KEITIKLIAI (ENKODERIAI).....	15
3. LINIJINIO ABSOLIUTINIO POSLINKIO ALGORITMŲ MODELIAVIMAS	16
3.1. Modeliavimo tikslai	16
3.2. Nuskaitymo klaidų klasifikavimas	16
3.2.1. Pavienės klaidos	16
3.2.2. Grupinės klaidos.....	17
3.2.3. Klaidos atsirandančios dėl skalės ir optinio sensoriaus žingsnių pločio nesutapimo	17
3.3. Modeliavimo metodika	18
3.4. Modelio testavimas	18
3.4.1. Netikslaus pozicionavimo įtaka	19
3.4.2. Jutiklio pikselio ir liniuotės nesutapimo įtaka.....	21
3.5. Pozicijos nustatymo patikimumas naudojant pseudo atsitiktines sekas.....	23
3.5.1. Sekų generavimas.....	23
3.5.2. Koreliacinis algoritmas	24
3.5.3. Euklido atstumo algoritmas.....	28
3.5.4. Euklido ir koreliacijos algoritmų palyginimas.....	30
3.5.5. Inversinio LFSR (Linear Feedback Shift Register) generatoriaus algoritmas	35
3.5.6. Patobulintas inversinio LFSR generatoriaus algoritmas	36
4. LINIJINIO ABSOLIUTINIO POSLINKIO ALGORITMŲ REALIZAVIMAS	41

4.1.	Galimos aparatinės įrangos alternatyvos	41
4.2.	Mikroprocesorinės priemonės	41
4.3.	Programuojamos logikos priemonės	47
	IŠVADOS	49
	LITERATŪROS ŠALTINIAI	50

IVADAS

Šiuolaikinėje pramonėje žmonių darbas keičiamas robotų, siekiant padidinti gamybos našumą, bei sumažinti gaminamo produkto savikainą. Robotai gali atlikti įvairias užduotis, pasikartojančias ar pavienes, daug didesniu tikslumu nei tai padarytų žmonės. Pagrindinė pramonės robotų, taip pat gamybos staklių bei mašinų tikslaus ir precizinio veikimo sąlyga - tikslus valdomo objekto pozicijos žinojimas. Kaip pavyzdį paimkime frezavimo stakles. Frezavimo staklių užduotis yra pašalinti frezuojamos medžiagos plotus tiksliai projektuotojo numatytose vietose. Jei nebus užtikrinama tikslus frezos veikimas ir jos padėties nustatymas, gaminama detalė gali būti pagaminta su broku, neatitiktį projektuotojo užsibrėžtų reikalavimų. Dėl šios priežasties gamybos mašinų tikslus padėties nustatymas yra viena svarbiausių užduočių. Kitas svarbus veiksnys siekiant didinti gamybos produktyvumą - tai padėties nustatymo greitaveika. Jei gamybos mašinų padėtis nustatoma ne pakankamai greitai, gali atsirasti paklaidos, taip pat gamybos procesas užtrunka ilgiau.

Šio darbo tikslas yra ištirti gamybos mašinose naudojamus padėties nustatymo būdus ir juose taikomus padėties aptikimo algoritmus. Atlikti greitaveikos ir patikimumo tyrimus bei palyginti juos tarpusavyje, iškeliant kiekvieno algoritmo plusus ir minusus tam, kad galutinis vartotojas galėtų pasirinkti, kuris algoritmas yra tinkamiausias jo kuriamai sistemai.

Darbo tikslas: pasiūlyti ir ištirti algoritmus, skirtus absoliutinės padėties nustatymui, panaudojant pseudo atsitiktinę seką, užneštą ant keitiklio liniuotės, ir nustatyti jų privalumus bei trūkumus, realizuojant ribotų resursų įterptinėje sistemoje.

Tikslui pasiekti išskelti šie uždaviniai:

1. Apžvelgti linijinių keitiklių konstrukcinius sprendimus ir identifikuoti įtakos faktorius, kurie atlieka trukdantį poveikį, nustatant absoliutinę keitiklio padėtį;
2. Sukurti imitacinį modelį, kuris leistų generuoti optinio keitiklio signalus, esant užduoties trukdantiems faktoriams (pvz., pavienės ir grupinės klaidos dėl dulkių ant liniuotės, klaidos dėl nevienodo optinio jutiklio elementų apšvietimo ir t.t.);
3. Parinkti algoritmus absoliutinės padėties nustatymui, juos realizuoti modeliavimo aplinkoje Matlab ir Monte Karlo metodu ištirti jų atsparumą trukdantiems faktoriams. Pasiūlyti algoritmų modifikavimą ir adaptavimą atsparumo padidinimui bei tinkamumui realizuoti ribotų resursų įterptinėmis sistemomis
4. Įvertinti parinktų algoritmų greitaveiką bei našumo poreikius juos realizuojančiam elektroniniam įrenginiui.

1. ABSOLIUČIOJO ATSTUMO MATAVIMŲ METODŲ APŽVALGA

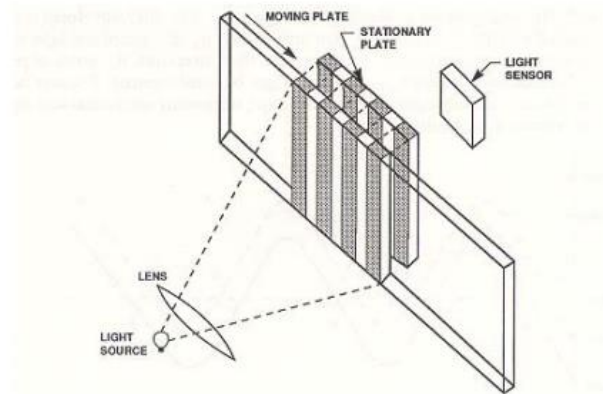
1.1. Enkoderių rūšys

Pramonėje naudojami enkoderiai skirstomi į du tipus: inkrementinius ir absoliučiuosius. Pagal veikimo principą jie skirstomi į dvi kategorijas: optinius ir magnetinius. Padėties kodavimas abiejose technologijose gali būti vienodas, o pagrindinis jų skirtumas tai, kad magnetinio enkoderio diske arba liniuotėje informacija koduojama skirtingai įmagnetinant tam tikras disko/liniuotės vietas, arba padarant fizinius įpjovimus. Skirtingas įmagnetinimas arba fiziniai įpjovimai keičia magnetinio lauko savybes, taip yra atskiriami loginiai nuliai ir vienetai. Optinio enkoderio veikimo principas paprastesnis, taip pat šiems enkoderiams būdinga galimybė išgauti didesnę skiriamąją gebą. Optinio enkoderio veikimas pagrįstas šviesos impulsų konvertavimu į elektrinius. Dažniausiai vienoje liniuotės/disko pusėje yra šviesos šaltinis, kitoje detektorius. Pats diskas/liniuotė suskirstytas į šviesai laidžias ir nelaidžias sekcijas, kurios reprezentuoja loginius vienetus ir nulius. Dėl pavaizdavimo ir veikimo principo paprastumo toliau darbe bus nagrinėjami tik optiniai enkoderiai, nors visi toliau aprašyti kodavimo būdai gali būti pritaikyti ir magnetiniuose enkoderiuose.

1.2. Inkrementiniai enkoderiai

Prieš tai minėta, kad enkoderiai pagal padėties nustatymo principą skirstomi į inkrementinius ir absoliučiuosius. Inkrementinio enkoderio veikimo principas paremtas šviesos impulsų skaičiavimu. Diskas arba liniuotė yra suskirstytas į sekcijas pasitelkus šviesai nepralaidžias juosteles. Šios juostelės viena nuo kitos yra nutolusios per vienodą atstumą – d , dažniausiai pačių juostelių plotis būna lygus atstumui - d . Liniuotei arba diskui sukantis šviesos šaltinį blokuoja periodiškai besikeičiančios šviesai pralaidžios ir nepralaidžios disko/liniuotės sekcijos, šviesos impulsai patenka į detektorius, kuriame generuojami elektriniai impulsai. Žinant disko/liniuotės parametrus ir suskaičiavus impulsus galima apskaičiuoti liniuotės arba disko padėtį pradinio atskaitos taško atžvilgiu. Taip pat išmatavus impulsų periodą galima nustatyti disko arba liniuotės judėjimo greitį. Liniuotės atveju, inkrementinio enkoderio skiriamoji geba nusakoma šviesai nelaidžių juostelių kiekiu, tam tikrame ilgio vienete. Skiriamoji geba enkoderio diske nusakoma bendru šviesai nelaidžių juostelių skaičiumi. Dažniausiai vietoj vienos nuskaitymo galvutės naudojamos dvi. Naudojant dvi nuskaitymo galvutes, antroji galvutė pozicionuojama esant per pusę liniuotės žingsnio[2]. Taip enkoderio skiriamoji geba padvigubinama, taip pat naudojant dvi skenavimo galvutes įmanoma nustatyti judėjimo kryptį. Kitas skiriamosios gebos pagerinimo būdas

– interpoliacija, kai panaudojamas antras toks pat enkoderio fragmentas [1]. Enkoderio fragmentas yra tarp enkoderio disko/liniuotės ir detektoriaus. Diskui judant jo šviesai nelaidžios juostelės atsiduria virš enkoderio fragmento, šviesai laidžių juostelių. Taip mažinamas šviesai laidas tarpas, kol jis visai išnyksta, o diskui toliau judant jis vėl pradeda didėti, taip gaunamas trikampio formos signalas. Galima teigti, kad šviesai laidas tarpas, taip pat ir gauto signalo amplitudė, yra tiesiogiai proporcingas disko/liniuotės ir enkoderio fragmento tarpusavio padėčiai. Tokio enkoderio pavyzdys pateiktas (1.1) paveiksle.

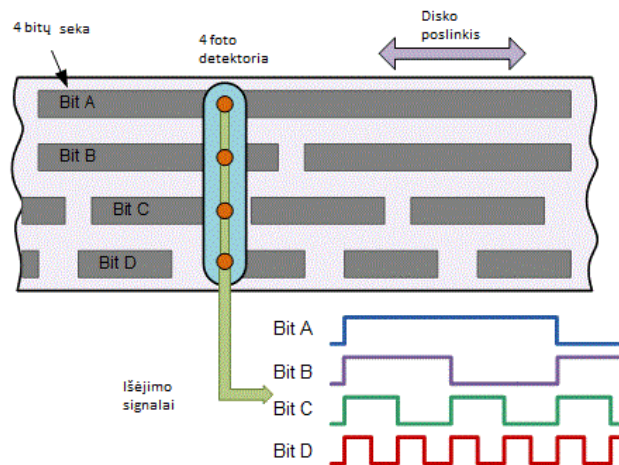


1.1 pav. Klasikinio absoliučiojo atstumo (postūmio) enkoderio pavyzdys

Pagrindinis inkrementinio enkoderio privalumas - tai paprasta konstrukcija ir galimybė nustatyti judėjimo greitį, tačiau jis turi vieną didžiulį trūkumą, kuris neleidžia jo panaudoti daugelyje situacijų. Inkrementinis enkoderis negali nusakyti tikslios padėties po maitinimo nutraukimo. Kaskart nutrūkus maitinimui ir jam atsistačius, prietaisas privalo būti grąžinamas į nulinę padėtį tam, kad vėliau galėtų būti tiksliai nusakyta jo padėtis. Dėl šios priežasties inkrementinio enkoderio panaudojimas yra negalimas tokiose vietose, kur matuojamo objekto grąžinimas yra negalimas. Taip pat dėl netinkamų eksploataavimo sąlygų, purvo, elektrinių trukdžių kai kurie impulsai gali likti neužfiksuoti, dėl šios priežasties skaičiuojant impulsus atsiranda paklaidų.

1.3. Absolūtieji enkoderiai

Dėl (1.2) skyriuje išvardytų inkrementinio enkoderio trūkumų plačiau naudojamas absoliutusias enkoderis. Pagrindinis absoliučiojo enkoderio skirtumas nuo inkrementinio enkoderio tai, kad net ir nutraukus maitinimą, ir jį atnaujinus, objekto padėtis gali būti nusakoma iškart, taip pat nebelieka paklaidos dėl impulsų praradimo. Klasikinis absoliučiojo enkoderio realizavimas pavaizduotas (1.2) paveiksle.



1.2 pav. - Klasikinio absoliučiojo enkoderio pavyzdys

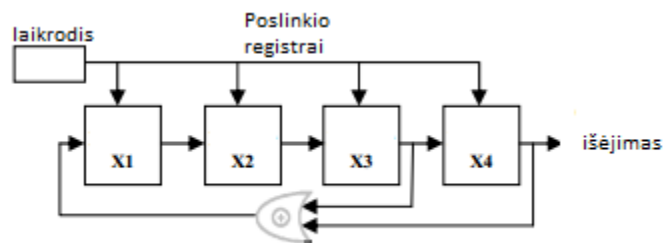
(1.2) paveiksle pavaizduotas absoliutinis enkoderis, kurio realizavimas įgyvendinamas panaudojant kelis lygiagrečius takelius. Kiekvienas takelis, kaip ir inkrementinio enkoderio atveju, suskirstytas į šviesai laidžias ir nelaidžias sekcijas, kurios savo ruožtu simbolizuoja loginius nulius ir vienetus. Detektorius vienu metu nuskaityto toje pozicijoje esančius takelius ir pagal tai, koks kodas gaunamas, yra nustatoma tiksli detektoriaus vieta disko arba liniuotės atžvilgiu. Dėl šios savybės, absoliutinis enkoderis, net ir nutrūkus maitinimui bei jį vėl atstačius, gali nuskaityti tikslią poziciją, nepakeitus detektoriaus ir liniuotės/disko tarpusavio padėties. Tačiau toks absoliutinio enkoderio dizainas kelia dvi pagrindines problemas. Pirmoji - siekiant gauti didesnę skiriamąją gebą reikalingas didesnis takelių skaičius, taip pat ir detektorių. Kitas trūkumas tai, kad vienetinio kodavimo principas nėra apsaugotas nuo neteisingo kodo interpretavimo, kai du ar daugiau bitų keičia savo reikšmę vienu metu. Su pastarąja problema kovojama panaudojant Grėjaus kodą [3], šio kodo kiekvienas paskesnis skaitmuo nuo praėjusio visada skiriasi tik vienu bitu. Dėl šios priežasties problema dėl neteisingo kodo interpretavimo išnyksta, bet atsiranda būtinybė atlikti Grėjaus kodo konvertavimą į vienetinį. Tačiau Grėjaus kodo panaudojimas neišsprendžia pagrindinės problemos, naudojant vienetinį kodavimą arba Grėjaus kodą abiem atvejais, siekiant gauti didesnę skiriamąją gebą, reikalingą pridėti papildomus kodo takelius. Šitoks skiriamosios gebos didinimas savo ruožtu padidina visą pozicijos nustatymo sistemos dydį, kas dažniausiai yra nepageidautinas reiškinys.

1.4. Nuoseklus kodavimo būdai

Yra keletas būdų, kai enkoderio skiriamoji geba yra didinama, tačiau papildomi kodo takeliai nepridedami. Vienas iš jų yra nuoseklusis kodavimas [4]. Koduojant nuosekliai kiekvienas kodo bitas nusakomas tam tikro ilgio šviesai pralaidžia arba nepralaidžia juosta, pavyzdžiui, pirmojo bito juostelės plotis 4 mm, antrojo - 6 mm, trečiojo 8 mm ir taip toliau. Pagrindinis šitokio metodo

trūkumas yra kodavimo skiriamosios gebos sumažėjimas, nes vienas padėtis kodo skaičius užima daug vietos. Tarkime padėtis koduojama 4 bitais, 1 bito plotis 4mm 2- 6mm, 3- 8mm, 4-12mm, bendras vieną padėtį nusakantis kodo ilgis tampa 30mm, kai analogiškas enkoderis koduojamas daugelio takelių būdu lieka tie patys 4 mm. Situacijose, kur reikalingas didelis tikslumas, šitoks kodavimo būdas netinka, tačiau toks kodavimo būdas leidžia išgauti ganėtinai ilgą kelių šimtų metrų kodo takelį [4], priklausomai nuo panaudotų bitų kiekio ir jų matmenų. Šitokia savybė gali praversti tokiais atvejais, kai labai didelis tikslumas nėra reikalaujamas. Pavyzdžiui, automatizuotuose logistikos sandėliuose, kur robotų nukeliamajami atstumai iki krovinio gali siekti šimtus metrų.

Siekiant išlaikyti didelę skiriamąją gebą ir nepanaudoti papildomų kodo takelių, naudojama pseudo atsitiktinė seka. Ši seka sugeneruojama pasinaudojant postūmio registru su atgaliniu grįžtamu ryšiu [5, 6], tokio sekos generatoriaus pavyzdys pateiktas (1.3) paveiksle.



1.3 pav. Pseudo atsitiktinės sekos generatoriaus pavyzdys

Atsitiktinės sekos generatoriaus fizinė struktūra gali būti nusakyta pagal polinominę lygtį [6], (1.3) paveiksle pavaizduoto sekos generatoriaus lygtis užrašyta (1) formulėje, joje X tai poslinkio registre esančio bito reikšmė, o laipsnis nurodo bito vietą registre.

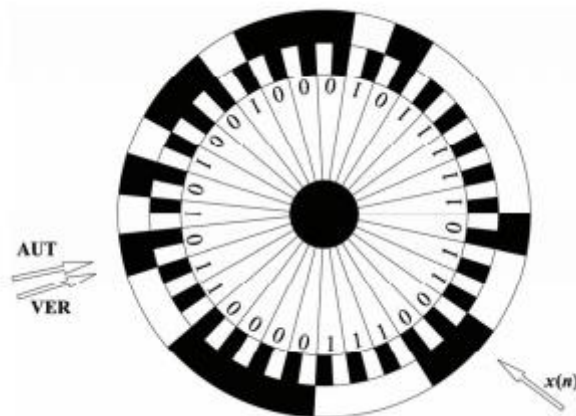
$$X^4 + X^3 + 1 \quad (1)$$

Šitokio generatoriaus sugeneruotai sekai būdinga tai, kad, pavyzdžiui, 4 bitų generatoriaus sugeneruotoje sekoje paėmus bet kokią 4 bitų dydžio eilę ji bus unikali [5]. Taip pat ši seka pasižymi periodiškumu; pasiekusi maksimalią reikšmę seka ima kartotis iš pradžių [7]. Maksimalią seką galima gauti generatorių aprašius pirmine polinominė lygtimi, pavyzdžiui, 16 bitų generatoriaus sugeneruotą seką, generatoriaus polinominė lygtis pateikta (2) lygtyje, pasikartojimo periodas bus lygus 65535 reikšmėms, šis skaičius gaunamas pagal (3) lygtį, čia n - tai maksimalus bitų skaičius.

$$X^{16} + X^{14} + X^{13} + X^{11} + 1 \quad (2)$$

$$2^n - 1 \quad (3)$$

Dėl pirmosios savybės pseudo atsitiktinė seka yra plačiai naudojama absoliutiniuose enkoderiuose, nes ji leidžia išlaikyti tokią pačią skiriamąją gebą kaip ir inkrementinio enkoderio atveju, taip pat norint padidinti skiriamąją gebą nereikia pridėti daugiau kodo takelių. Tačiau šio kodo panaudojimas turi ir savų trūkumų. Vienas iš jų, tai kodo nuskaitymas. Kadangi pseudo atsitiktinėje sekoje vienas po kito gali eiti keletas loginių vienetų arba nulių, jų tarpusavio padėties išskirimas tampa sudėtinga užduotimi. Paprasčiausias būdas šiai problemai išspręsti, tai sinchronizacijos takelio panaudojimas [8 ,10]. Sinchronizavimo takelis paklojamas šalia pseudo atsitiktinės sekos takelio, jis suskirstytas tokiu pačiu principu kaip ir klasikinio inkrementinio enkoderio takeliai, šviesai nelaidžių ir laidžių juostelių pločiai lygūs pseudo atsitiktinės sekos vieno bito ilgiui. Kadangi sinchronizacijos takelio bitai periodiškai kinta, pagal jų generuojamus šviesos impulsus galima lengvai nustatyti pseudo atsitiktinės sekos bitų pokyčius. Pagal pseudo atsitiktinę seką užkoduotas diskas su sinchronizavimo takeliu pavaizduotas (1.4) paveiksle



1.4 pav. Pseudo atsitiktinės sekos enkoderio pavyzdys

Taip pat siekiant padidinti kodo nuskaitymo patikimumą, lygiagrečiai pseudo atsitiktinės sekos takelio pridėdamas lygiai tokios pačios sekos kitas takelis, tačiau jo bitų žymėjimas yra atvirkščias, šitoks nuskaitymo būdas vadinamas diferenciniu [8]. Kitas būdas padidinti nuskaitymo patikimumą, tai panaudoti mančesterio kodavimą [9]. Koduojant seką Mančesterio kodu, loginiai "0" pakeičiami seka "01", o loginiai "1" - "10". Šitokiu būdu padidinamas perėjimų iš loginio "1" į loginį "0" skaičius, dėl ko galime tiksliau sinchronizuoti kodo nuskaitymą.

Kitas pseudo atsitiktinės sekos enkoderio trūkumas – būtinybė nusakyti nulinę padėtį po sumontavimo. Literatūros šaltiniuose [11] ir [12] pateikiamas paprastas algoritmas šiai problemai spręsti. Jo principas paremtas tuo, kad nuskaitytas kodo segmentas iš pseudo atsitiktinės sekos gali būti priskirtas pradinei sekos reikšmei. Algoritmo vykdymo pradžioje nuskaitytas kodo segmentas, jis įkeliamas į pseudo atsitiktinės sekos generatorių, toliau atliekamas kodo teisingumo tikrinimas.

Paeiliui yra nuskaitoma po vieną kodo bitą ir lyginama su gautu iš generatoriaus. Šitoks tikrinimas atliekamas iš anksto nustatytam bitų skaičiui, jei visi bitai sutapo, iš pat pradžių nuskaityta kodo reikšmė priskiriama pradinei pozicijai ir išsaugoma atmintyje. Jei aptinkamas nesutapimas, ciklas pradedamas iš pradžių.

Egzistuojantys nuskaitymo būdai gali būti suskirstyti į tris kategorijas - nuoseklus, lygiagretus ir mišrus. Nuoseklūs būdai aprašyti [13], [5], [14], [15], [16], [17] literatūros šaltiniuose. Konvertavimo principas, nuoseklioju būdu, paremtas generatoriaus sugeneruotu kodo reikšmių skaičiavimu. Generatoriaus sugeneruotos kodo reikšmės lyginamos su galutine sekos reikšme kol gaunamas kodas lygus nuskaitytam. Jei pasirenkamas būdas naudoti atgalinį generavimą, nuskaitytas kodas yra perduodamas į generatorių kaip pradinė reikšmė. Toliau generatorius yra taktuojamas iki kada gaunama kodo reikšmė atitinkanti nulinę padėtį. Kodų lyginimo operacijos atliekamos komparatoriumi arba pasitelkiant koreliaciją. Šio metodo privalumai tai maži reikalavimai atminčiai, tačiau toks paieškos būdas ganėtinai lėtas.

Lygiagretus nuskaitymo metodas nuo nuoseklaus skiriasi tuo, kad yra paremtas pseudo atsitiktinės sekos išsaugojimu atmintyje [5], [17]. Sugeneruota ir išsaugota atmintyje pseudo atsitiktinė seka yra identiška sekai, kuria koduojama liniuotė arba diskas. Pozicijos nustatymo metu nuo liniuotės arba disko yra nuskaitomas kodas, kuris vėliau koreliacijos metodu arba panaudojant komparatorių yra lyginamas su išsaugota seka. Radus kodo atitikmenį sekoje, išsaugoma atitikmens pozicija, pagal ją galima nustatyti kokioje pozicijoje detektoriaus atžvilgiu yra liniuotė arba diskas. Mišrus konvertavimo būdas, tai kai siekiant pakeisti pseudo atsitiktinės sekos kodo reikšmę į natūrinį kodą panaudojamas nuoseklūs ir lygiagretūs būdai [5], [17].

Mišraus konvertavimo atveju atmintyje yra saugomi pseudo atsitiktinės sekos fragmentai, kitaip tariant žymos [5]. Detektoriumi nuskaitomas padėties kodas, kuris yra sulyginimas su išsaugotomis žymomis, jei sutapimas yra neaptinkamas, nuskaitytas kodas, sekos generatoriumi, perstumiamas per vieną padėtį pirmyn arba atgal, taip kad simbolizuotų gretimą padėtį. Naujai sugeneruotas kodas lyginamas su išsaugotomis žymomis, jei vėl nėra randama sutapimų, viskas kartojama iš naujo iki kol sutapimas aptinkamas. Aptikus atitikmenį su išsaugota žyma, nustatoma jos padėtis sekoje. Tikroji padėties reikšmė randama atėmus arba pridėjus, priklausomai į kurią pusę buvo stumiami bitai, skaičių lygų atliktų perstūmimų skaičiui. Literatūros šaltinyje [17] aprašytas mišrus nuskaitymo būdas paremtas $2^{(N/2)} \times 2^{(N/2)}$ matricos sudarymu, N tai skaičius generatoriaus su kuriuo buvo generuojama bitų seka. Visa matrica paeiliui užpildoma skaičiais iš pseudo atsitiktinės sekos, kiekvienai matricos celei priskiriamas eilės numeris, kuris atitinka toje celėje esančio kodo padėtį pseudo atsitiktinėje sekoje, paskutinis matricos stulpelis išsaugomas. Pagrindinis skirtumas nuo prieš tai minėto algoritmo tai, kad čia stulpelis surūšiuojamas taip, kad būtų galima pritaikyti greitąjį paieškos algoritmą. Toliau kaip ir [5] šaltinyje aprašytame algoritme

yra ieškomas sutapimas išsaugotame stulpelyje, jo neradus kodo reikšmė yra perstumiama per vieną žingsnį ir vėl atliekama paieška.

2. LINIJINIO ABSOLIUTINIO POSLINKIO KEITIKLIAI (ENKODERIAI)

Rinkoje plačiai naudojami absoliutiniai enkoderiai paremti ne tik klasikiniu principu, kai naudojama keletą lygiagrečių takelių, kurių kiekvienas simbolizuoja atskirą kodo bitą, bet ir paremtų pseudo atsitiktine seka. (1) Lentelėje pateikti techniniai duomenys realių absoliutinių linijinių enkoderių. Šioje lentelėje matome tokių gamintojų kaip Heidenhain, Acu-Rite, Celera, Renishaw, GSI, Newall gaminių techninius duomenis.

Lentelė 1. Rinkoje esančių enkoderių lentelė

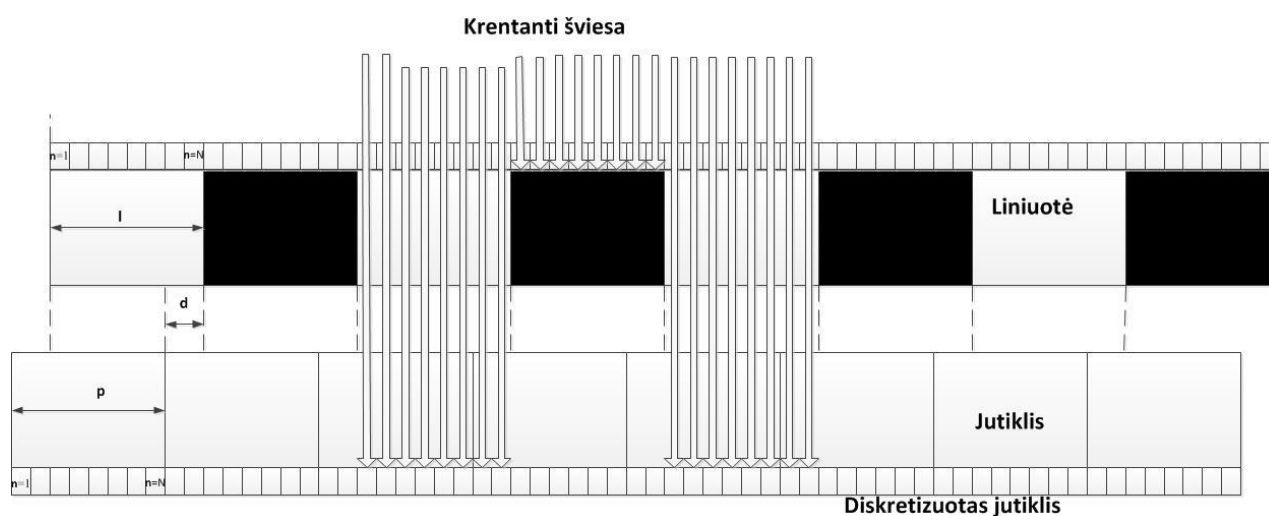
Gamintojas	Modelis	Maksimalus ilgis	Maksimali rezoliucija
ACU-RITE	SENC 150	3048 mm	20 um
	SENC 150	508 mm	
Heidenhain	LC415	2040 mm	20 um
	LC485		
	LC495		
	LC115	42420 mm	20 um
	LC185		
	LC195		
	LC211	28040mm	40 um
	LC281		
	LC291		
Celera	MicroE		40 um
Renishaw		5000 mm	
GSI	Mercury 1000		20 um

Nors rinkoje absoliutinę poziciją nustatančių prietaisų yra, tačiau tikslus algoritmai, pagal kuriuos nustatoma pozicija nėra atskleidžiami, dėl šios priežasties pagrindinis darbo tikslas ištirti žinomus padėties algoritmus, taip pat pasiūlyti jų optimizavimo arba tobulinimo variantus.

3. LINIJINIO ABSOLIUTINIO POSLINKIO ALGORITMŲ MODELIAVIMAS

3.1. Modeliavimo tikslai

Šioje darbo dalyje bus kuriamas ir testuojamas iš optinio detektoriaus gaunamo signalo modelis. Šitokio modelio pagrindinis poreikis - turėti didesnę testinių signalų variaciją. Turimų realių signalų įrašų kiekis ribotas, todėl neturint realios aparatūros galimi tyrimai su esamomis sekomis tampa riboti. Taip pat turint modelį atsiranda galimybė į signalą įterpti įvairias paklaidas ir klaidas, tai padaryti su realia įranga gali būti sudėtinga. Turint modelį galima modeliuoti įvairaus matmenų sensorius bei liniuotes.



3.1 pav. Realizuoto modelio principinė schema

3.2. Nuskaitymo klaidų klasifikavimas

3.2.1. Pavienės klaidos

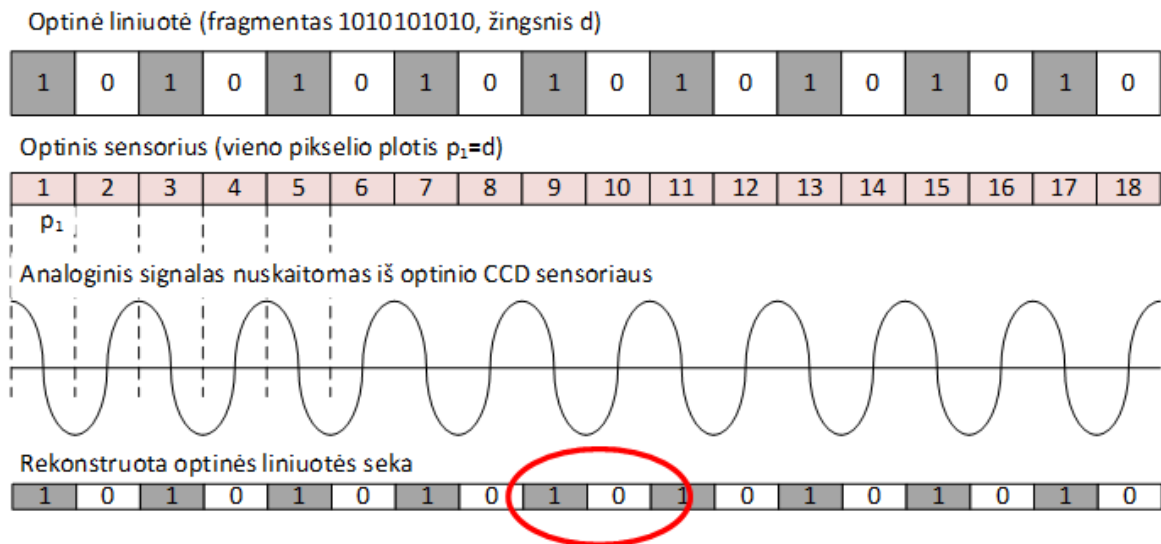
Pavienės paklaidos, tai neteisingai nuskaityti arba interpretuoti pavieniai bitai. Šitokių klaidų atsiradimą dažniausiai įtakoja optinės liniuotės nešvarumai, dulkės. Nukritusios ant liniuotės dulkės uždengia tarpus pro kuriuos turėtų kristi šviesa. Nešvarumo paveiktoje liniuotės vietoje turėtų būti identifiukuotas loginis „1“, tačiau dėl uždengtos šviesos į jutiklį, šviesa nepatenka ir jutiklio išėjimo signale ši konkreiti vieta interpretuojama kaip loginis „0“.

3.2.2. Grupinės klaidos

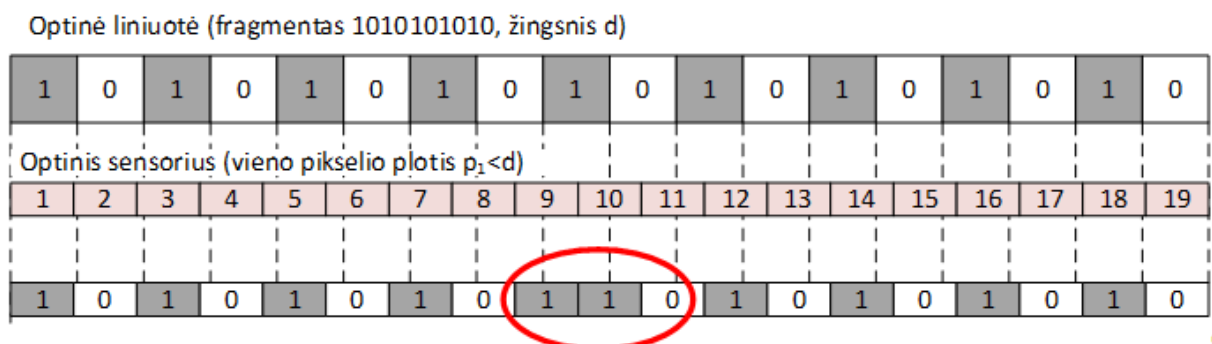
Grupinės klaidos dažniausiai pasitaiko atsitiktinai. Pagrindinis skirtumas, kad šios klaidos įtakoja ne vieną bitą, o keletą bitų arba visą signalą iš karto. Tokių klaidų atsiradimas, kaip ir pavienių klaidų atveju, gali įtakoti tokie faktoriai kaip nešvarumai ant liniuotės, stambios dulkės, kurios uždengia ne vieną, o keletą bitų.

3.2.3. Klaidos atsirandančios dėl skalės ir optinio sensoriaus žingsnių pločio nesutapimo

Šios paklaidos pobūdis, skirtingai nei prieš tai aptartų paklaidų rūšių, atsiranda dėl optinės skalės tarpelių žingsnio ir optinio sensoriaus pikselio pločio nesutapimo.



3.2 pav. Sekos pavyzdys, kai žingsniai sutampa



3.3 pav. Sekos pavyzdys, kai žingsniai nesutampa

Jutiklio išėjime atsiranda klaida, kai jutiklio pikselio žingsnis yra mažesnis nei liniuotės žingsnis. Kadangi viena pikselio reikšmė atitinka vieną liniuotės reikšmę, esant pikseliui

trumpesniai nei liniuotės žingsnis, dviem pikseliams tenka vieno šviesa, kuri turėtų patekti tik į vieną pikselį. Liniuotės ir pikselio žingsnio nesutapimas parodytas paveiksle (3.2 ir 4.3). Šios paklaidos įtaka jutiklio signalui su kiekvienu pikseliu vis didėja, dėl šios priežasties galiausiai įvyksta neteisingas bito identifikavimas, kaip pavaizduota (3.3) paveiksle.

3.3. Modeliavimo metodika

Kuriamame modelyje modeliuojamas šviesos jutiklis, kurį sudaro šviesai jautrių elementų (taškų arba pikselių) eilė arba matrica. Netikslus jutiklio ir liniuotės išlygiavimas ($|d| > 0$, 4.1 paveikslas), arba pikselio pločio bei liniuotės tarpelio pločio nesutapimas (l ir p 4.1 paveikslas), nuskaitytame kode, gali įnešti papildomas klaidas.

Kuriamame modelyje imituojamas apšvietimo lygis, tenkantis kiekvienam sensoriaus pikseliui. Analoginio signalo reikšmė, atitinkanti vieno pikselio apšviestumą apskaičiuojama pagal (3) formulę.

$$E(k) = \sum_{n=(k*N)-N}^{N*k} I[n] \quad (3)$$

Šioje formulėje $E(k)$, tai vieno pikselio analoginė reikšmė, N – pikselio diskretizavimo narių skaičius, k – pikselio eilės numeris, $I[n]$ - diskretizuoto pikselio vienos išrankos apšviestumo reikšmė. Modelyje padaroma prielaida, kad diskretizuotame pikselyje į laukelį $I[n]$ šviesa patenka arba ne, todėl $I[n]$ reikšmė laikoma lygi 1 arba 0.

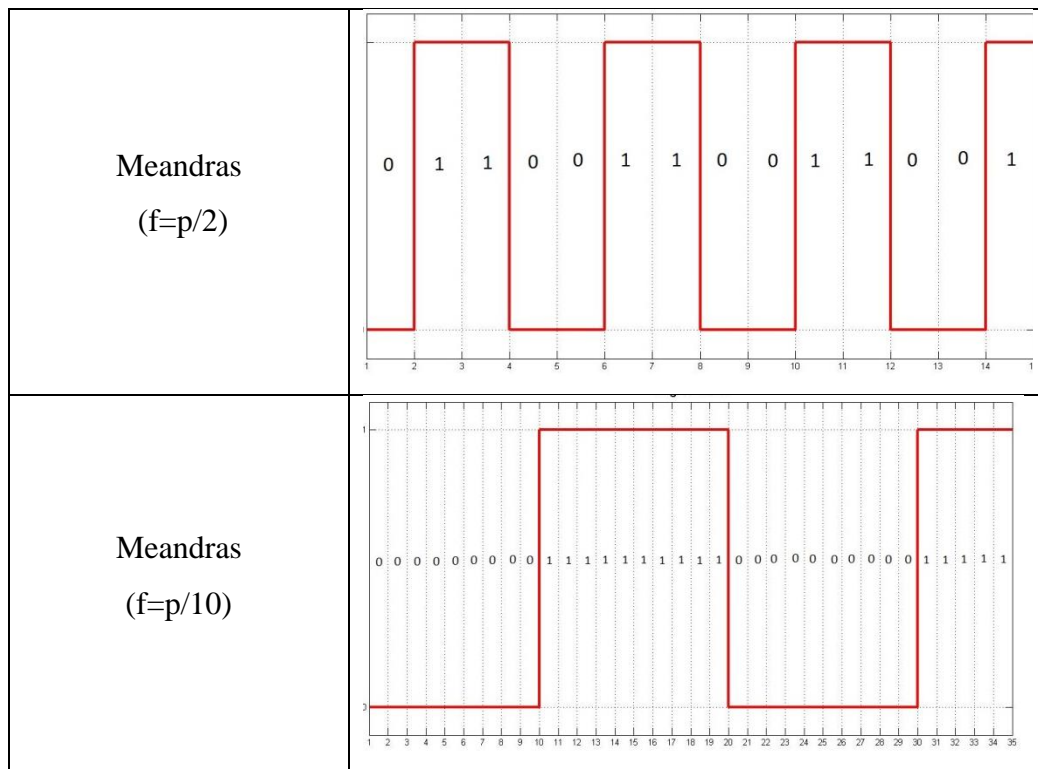
Modelyje galima keisti persistūmimo laipsnį (d), pikselio plotį (p), liniuotės gardelės žingsnį (l), pikselio diskretizavimą (n).

3.4. Modelio testavimas

Modelis testuojamas atliekant bandymus su keletu skirtingų testavimo sekų. Naudojamos sekos surašytos (2) lentelėje.

Lentelė 2. Testinių sekų lentelė

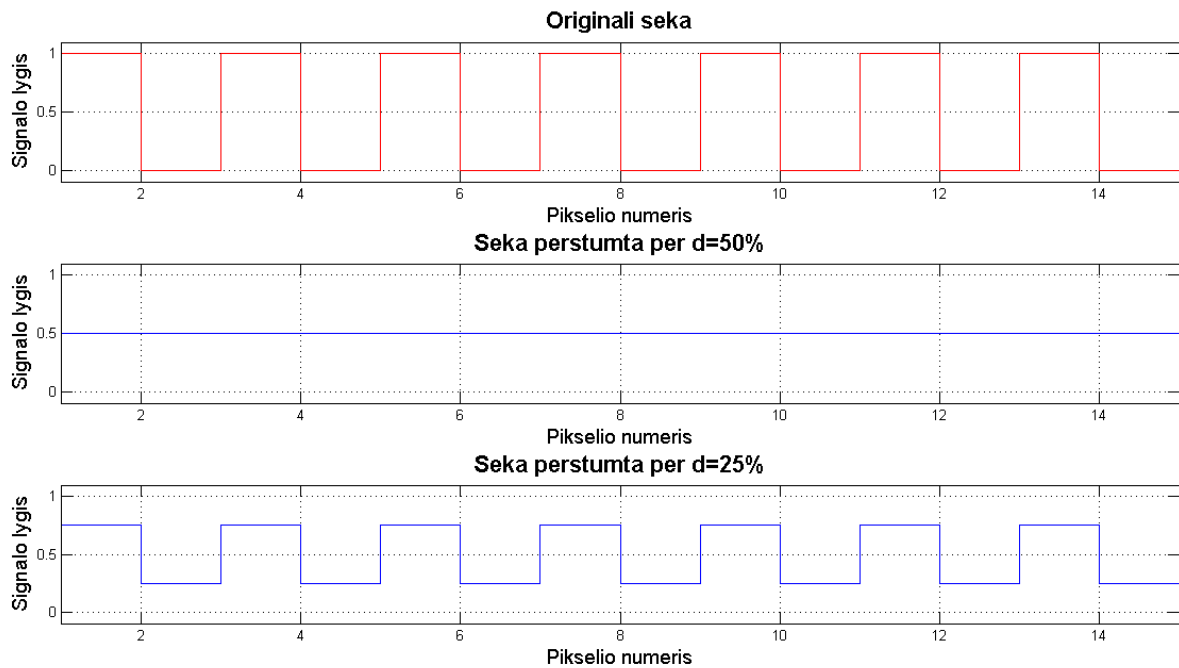
Testinė seka	Sekos grafinis atvaizdas
Meandras ($f=p$)	



3.4.1. Netikslaus pozicionavimo įtaka

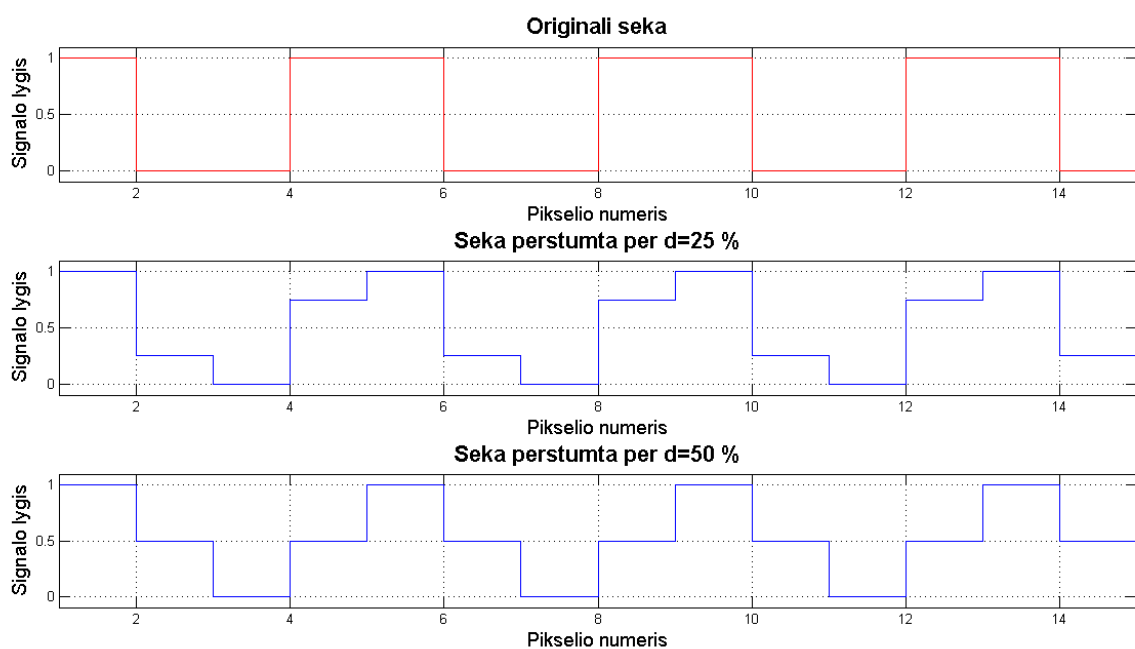
Pirmiausia atliekamas eksperimentas, kai šviesos jutiklio pikselio plotis ir liniuotės žingsnis sutampa ($p=l$). Atliekama tik netikslaus liniuotės pozicionavimo jutiklio atžvilgiu modeliavimas. Modelyje liniuotė pozicionuojama taip, kad pikselis ir liniuotės reikšmė nesutaptų per 25% ir 50%. Tai yra, parametras d sudaro 25% arba 50% parametro p reikšmės.

Gauti rezultatai pateikti paveiksluose (3.4). Juose x ašis atitinka modeliuojamo pikselio numerį, signalo lygis – signalo lygi jutiklio išėjime (1 – idealus loginis „1“, 0 – idealus loginis „0“).



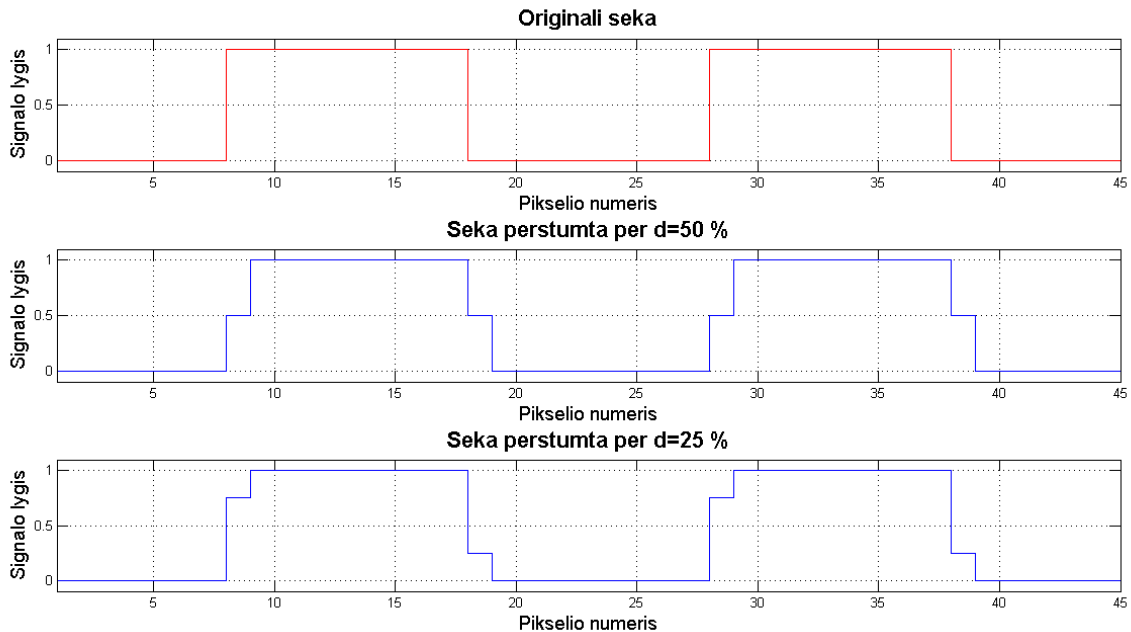
3.4 pav. Meandro ($f=p$) modeliuota seka esant pozicionavimo paklaidai

(3.4) paveiksle pavaizduotas meandro ($f=p$) signalas: viršuje pradinė seka, apačioje pavaizduojamas išėjimo signalas iš jutiklio, kai liniuotė pozicionuojama su 25% paklaida į dešinę pusę nuo tikslios padėties, viduryje su 50% paklaida. Taip pat atliktas eksperimentas, kai liniuotė parslenkama į kairę pusę atitinkamai 25 ir 50 % nuo tikslios pozicionavimo padėties. Šio eksperimento rezultatai identiški (3.4) paveiksle pavaizduotiems rezultatams. Iš paveikslo matyti, kad meandro pobūdžio sekos signalui pasiekus 50% pozicionavimo nesutapimą, nebeįmanoma nuspręsti ir rasti tinkamos sekos, nes išėjime gaunamas tolygus ir nekintantis signalas.



3.5 pav. Eksperimentas su meandro seka kai $f=p/2$

Kaip galime pastebėti iš (3.5) paveikslo esant 25% paklaidai, signalas nebesiekia maksimalios ir minimalios reikšmės, tačiau viršija pusės loginio lygio ribą, todėl spręsti apie jo teisingumą įmanoma. Kitokia situacija atsiranda esant 50% paklaidai, tokiu atveju, gretimi bitai kurių loginiai lygiai skiriasi, tampa tarpusavyje neišskiriami, nes išėjime esantis signalas yra ties pusę signalo lygio, todėl neįmanoma spręsti apie užfiksuoto bito tikrąją reikšmę.



3.6 pav. Eksperimentas su meandro seka kai $f=p/10$

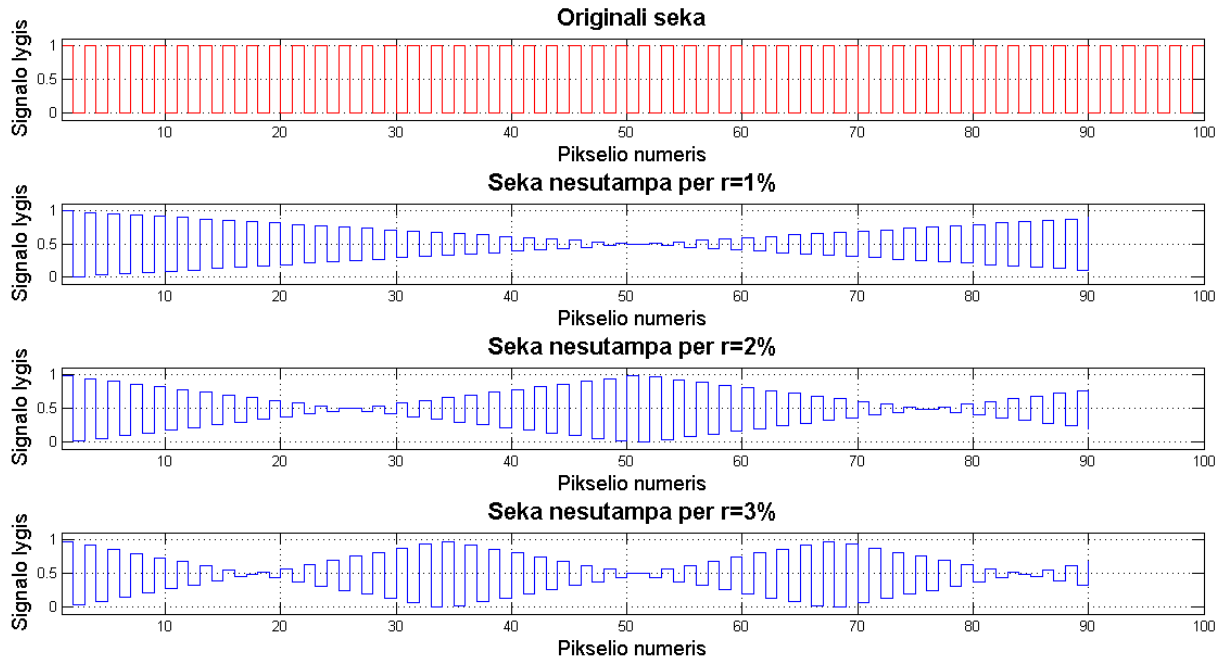
Toliau atliekamas eksperimentas, su meandro formos bitų seka, kurio dažnis $f=p/10$. Šioje sekoje kaip ir prieš tai darytuose eksperimentuose, ties 25% perstūmimo paklaida, dar galima išskirti bitus, tačiau esant 50% paklaidai, tokios galimybės jau nebelieka. Pirmasis bitas, kuris keičia originalioje sekoje loginį lygį, gautame signale tampa teisingai nebu nustatomu, nes signalo išėjimas yra ties 0.5 lygiu. Šią problemą būtų galima spręsti naudojant platesnio žingsnio liniuotę, taip, kad viename žingsnyje tilptų du ar daugiau pikselių, šitaip būtų galima eliminuoti atsiradusią netikslaus pozicionavimo paklaidą.

3.4.2. Jutiklio pikselio ir liniuotės nesutapimo įtaka

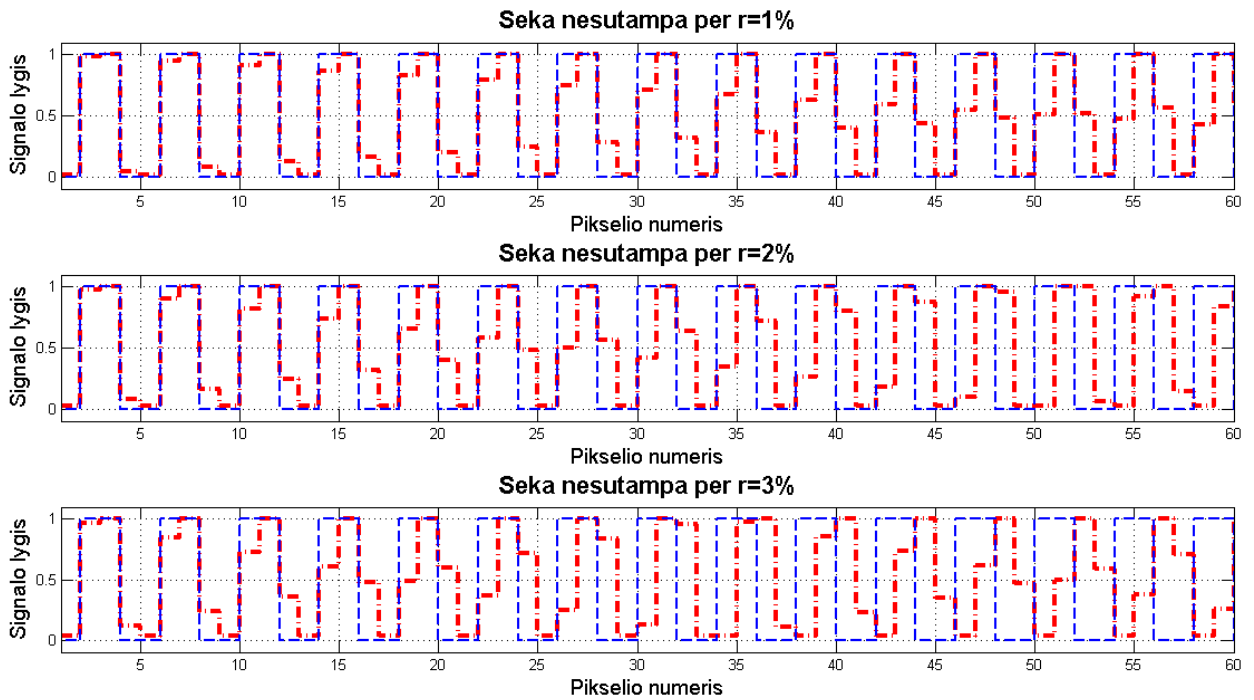
Toliau atliekamas eksperimentas kai jutiklio pikselių plotis nesutampa su liniuotės pločiu. Nesutapimas išreiškiamas procentais. Eksperimento metu naudojamas jutiklis, kurio pikselių plotis atitinkamai 1, 2 ir 3 procentais mažesnis nei liniuotės žingsnis (4) formulė.

$$r = \frac{(p-l)*100}{l} \quad (4)$$

Šioje formulėje r - tai nesutapimo lygis, p – jutiklio pikselio plotis, l – ant liniuotės esančio kodo žingsnio plotis.



3.7 pav. Signalų paklaida kai meandro (f=p) signalų žingsniai nesutampa



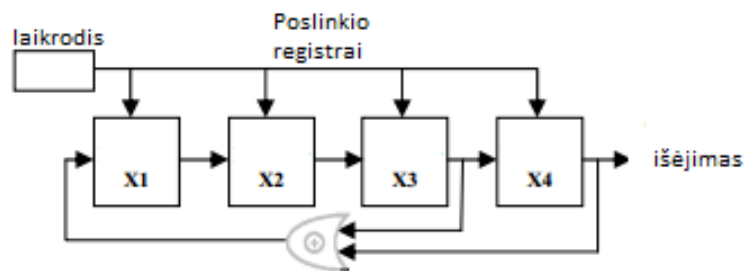
3.8 pav. Eksperimentas su meandro seka (f=p/2)

Kaip galima pastebėti iš paveikslų (3.8), esant nevienodam pikselio ir liniuotės žingsniui atsiranda adaptyvi paklaida. Ši paklaida geriausiai matoma (3.7) paveiksle, kai naudojamas meandro signalas, kurio dažnis $f=p$. Pirmasis pikselis užsibaigia greičiau nei liniuotės tarpas, todėl sekantis pikselis gauna nedidelę dalį apšviestumo, kuris turėjo tekti pirmajam pikseliui. Kadangi kiekvienas pikselis yra trumpesnis už liniuotės žingsnį, sekančiam pikseliui netinkamo apšviestumo dalis mažėja. Dėl šios priežasties paklaida didėja. Iš (3.8) paveikslo galima matyti, kad didėjant pikselio numeriui, ši paklaida įneša tokius netikslumus, dėl kurių teisingą bitą nustatyti tampa sunku arba neįmanoma.

3.5. Pozicijos nustatymo patikimumas naudojant pseudo atsitiktines sekas

3.5.1. Sekų generavimas

Šiame skyriuje sugeneruojamos pseudo atsitiktinės sekos, kurios bus reikalingos atliekant tolimesnius eksperimentus su padėties aptikimu. Sekos generavimas atliekamas panaudojus Fibionaci LFSR registrą, kurio principinė veikimo schema pateikta (3.9) paveiksle.



3.9 pav. Pseudo atsitiktinės sekos generatorius

LFSR registrai gali būti aprašyti polinominėmis lygtimis, (3.9) paveiksle pavaizduotas generatorius yra aprašomas (5) lygtimi.

$$X^4 + X^3 + 1 \quad (5)$$

Sekų generavimui pasitelkiama Matlab programinė įranga. Joje sukuriama programa galinti pagal pateiktus sekos parametrus: sekos ilgį, pradinę generatoriaus reikšmę, naudojamus polinomus, sugeneruoti pseudo atsitiktinę seką. Tolimesniems bandymams atlikti sugeneruojama pseudo atsitiktinė seka, kurios duomenys pateikti (3) lentelėje.

Lentelė 3. testinės sekos parametrai

Parametro pavadinimas	Reikšmė
Polinominė lygtis	$X^{17} + X^{14} + 1$
Generatoriaus bitų skaičius	17
Maksimalus periodo ilgis	$2^n - 1 = 131071$
Polinomų skaičius	2
Sekos ilgis	65535
Pradinė sėkla	1

Turint sugeneruotą bandomąją seką galima nagrinėti koreliaciją, Euklido atstumu bei LFSR generatoriais paremtus padėties aptikimo algoritmus.

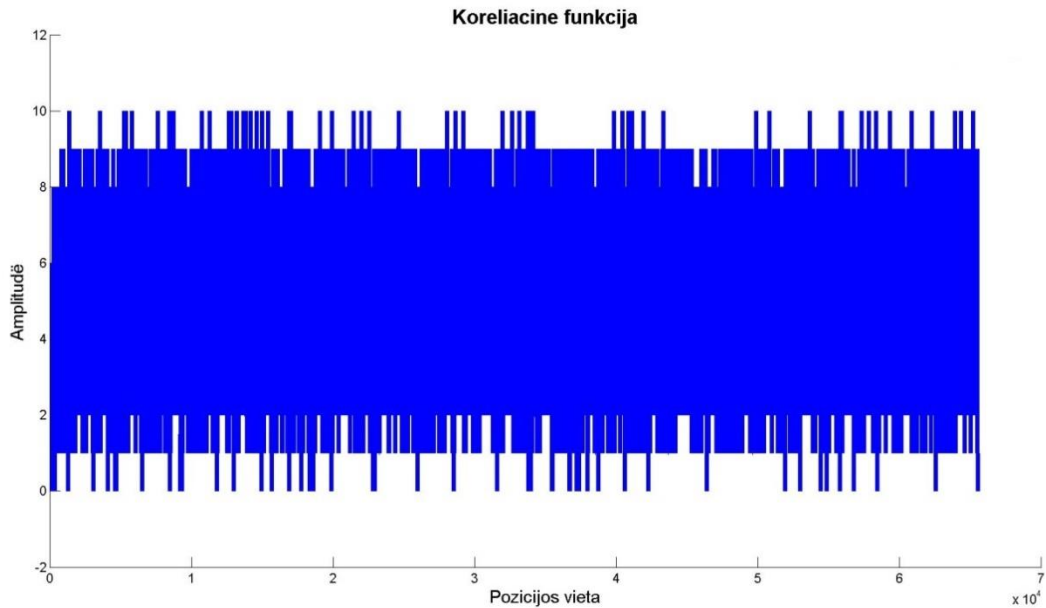
3.5.2. Koreliacinis algoritmas

Koreliacija tai funkcija, kuri ieško panašumų tarp dviejų skirtingų signalų. Kuo du lyginamieji signalai yra panašesni vienas į kitą, tuo apskaičiuotos koreliacijos funkcijos amplitudė turėtų būti didesnė. Vadinasi, nuskaitytam kodo segmentui, sutapus su sekos dalimis, koreliacijos funkcija turėtų būti maksimali, šio maksimumo vieta nurodytų nuskaityto kodo vietą sekoje. Koreliacijos funkcija apibrėžiama (6) formule.

$$S_{xy}(k) = \sum_{n=N}^N x(n)y(n - k) \quad (6)$$

Šioje formulėje S - tai gauta koreliacinė funkcija. N – tai atraminės sekos ilgis. Y – nuskaitytas sekos fragmentas. X – Atraminė seka. K - tai vėlinimo parametras. Maksimali koreliacinės funkcijos S vieta atitinka atraminėje sekoje ieškomo fragmento vietą.

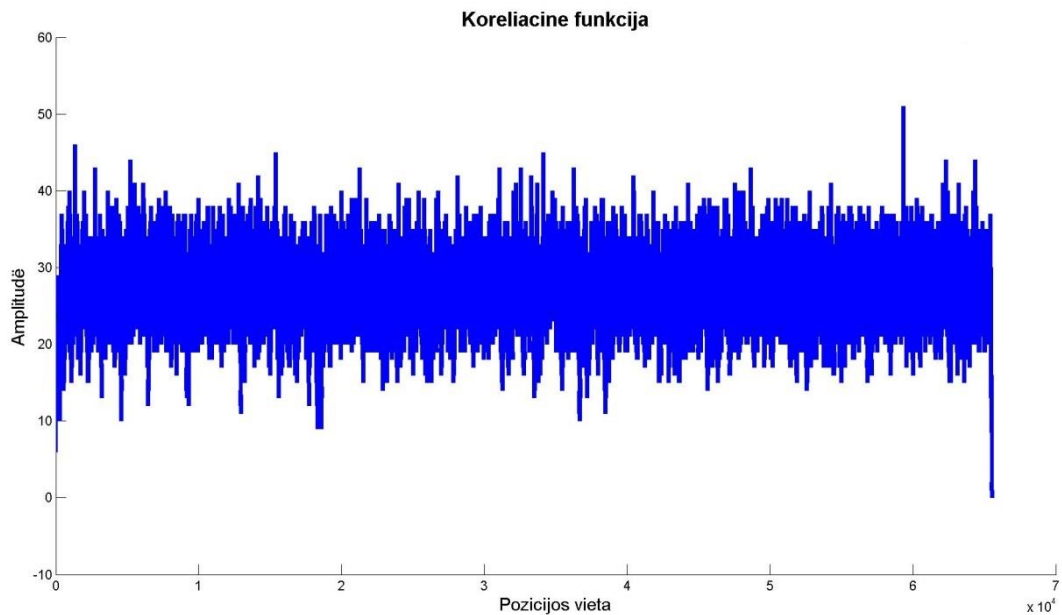
Pirmiausia atliekamas eksperimentas, kai nuskaityto kodo ilgis yra lygus generatoriaus bitų skaičiui. Šiuo konkrečiu atveju, nuskaityto kodo ilgis 17 bitų. Kodo nuskaitymo vieta pasirenkama atsitiktinai. Atlikus koreliacijos funkciją tarp nuskaityto kodo ir testuojamos sekos, gauta koreliacinė funkcija pateikta (3.10) paveiksle.



3.10 pav. Koreliacijos funkcija, kai nuskaityto kodo ilgis lygus generatoriaus ilgiui (17 bitų)

Kaip galime pastebėti iš (3.10) paveiksle pavaizduotos koreliacinės funkcijos, nematyti jokio akivaizdaus maksimumo. Kadangi nuskaitytas kodas buvo nukopijuotas iš testavimo sekos be jokių klaidų, galimybė, kad kodo nėra sekoje atkrenta.

Priežastis tokių rezultatų, tai per mažas nuskaityto kodo ilgis. Jeigu sekos ilgis būtų lygus maksimaliam periodui. Šioje sekoje galėtume rasti visus galimus septyniolikos bitų skaičius nuo 1 iki 131071. Dėl to, nuskaitytas 17 bitų kodas, turi daug į save panašių sekų, kurios skiriasi tik vienu arba dviem bitais. Kaip matome iš (3.10) paveikslo, tokio skirtumo nepakanka efektyviai išskirti nuskaityto kodo vietas. Toliau atliekamas eksperimentas, kai nuskaitytų bitų skaičius padidinamas iki 100 bitų. Gauta koreliacinė funkcija pavaizduota (3.11) paveiksle.



3.11 pav. Koreliacijos funkcija, kai nuskaityto kodo bitų ilgis lygus 100

Kaip galime pastebėti iš (3.11) paveikslo, padidinus nuskaitytų bitų skaičių nuo 17 iki 100, koreliacijos funkcijoje pastebimas ryškus maksimumas. Šio maksimumo vieta koreliacijos funkcijoje sutampa su ieškomo kodo vieta pseudo atsitiktinėje sekoje. Dėl šios priežasties galime teigti, kad turint pakankamai ilgą pseudo atsitiktinės sekos fragmentą, naudojant koreliacijos metodą galime tiksliai rasti jo padėtį toje sekoje.

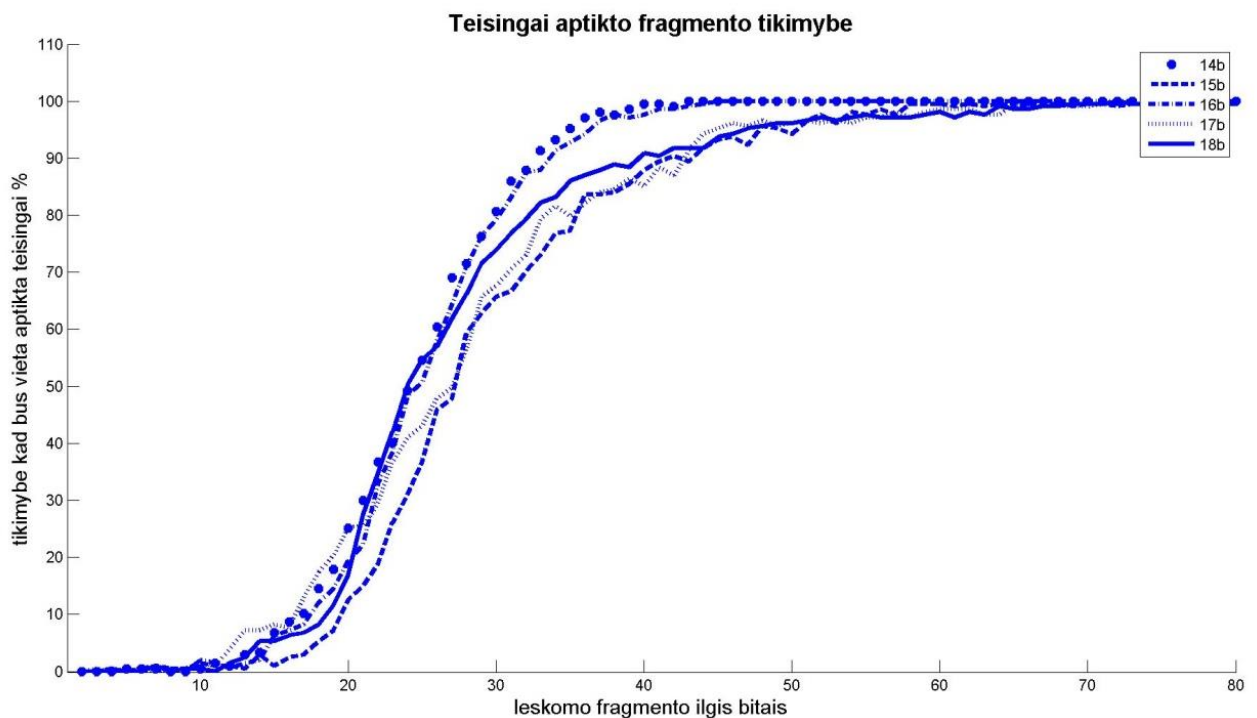
Toliau atliekamas eksperimentas siekiant nustatyti priklausomybę tarp koreliacijos būdų teisingai aptiktos pozicijos ir nuskaityto fragmento ilgio. Eksperimento metu imamos skirtingos sekos, naudojamų sekų duomenys pateikti (4) lentelėje. Fragmentai ieškomi tik iš pirmųjų 10 000 galimų fragmentų. Fragmento ilgis kinta nuo 2 bitų iki 500.

Lentelė 4 – Eksperimentuose naudojamų sekų duomenys

Sekos numeris	Generatoriaus bitų skaičius	Maksimalus periodo ilgis	Polinomų skaičius	Polinomu numeriai			
8	10	1023	2	10	7	-	-
9	11	2047	2	11	9	-	-
10	12	4095	4	12	6	4	1
11	13	8191	4	13	4	3	1
12	14	16383	4	14	5	3	1
13	15	32767	2	15	14	-	-
14	16	65535	4	16	15	13	4

15	17	131071	2	17	14	-	-
16	18	262143	2	18	11	-	-
17	19	524287	4	19	6	2	1
18	20	1048575	2	20	17	-	-
19	21	2097151	2	21	19	-	-
20	22	4194303	2	22	21	-	-
21	23	8388607	2	23	18	-	-
22	24	16777215	4	24	23	22	17
23	25	33554431	2	25	22	-	-
24	26	67108863	4	26	6	2	1
25	27	134217727	4	27	5	2	1
26	28	268435455	2	28	25	-	-
27	29	536870911	2	29	27	-	-
28	30	1073741823	4	30	6	4	1
29	31	2147483647	2	31	28	-	-

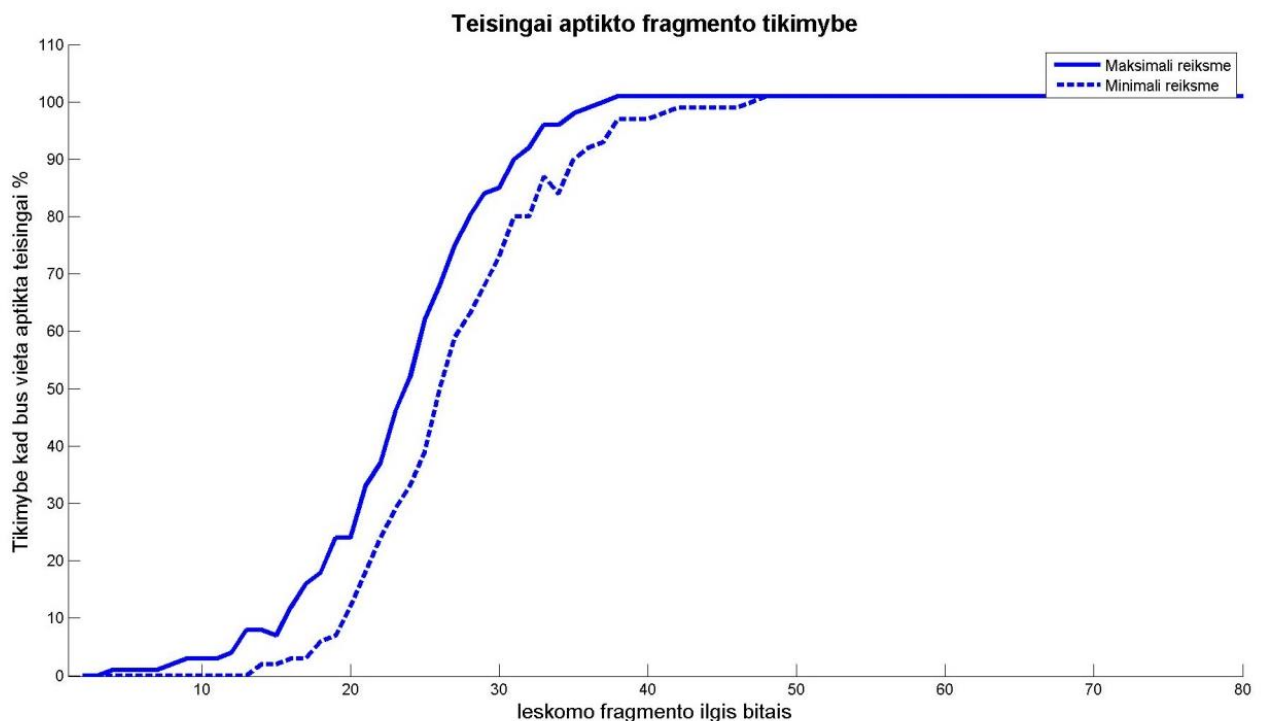
Gauti eksperimentų rezultatai pavaizduoti (3.12) paveiksle.



3.12 pav. Koreliacijos metodo vietos aptikimo tikimybės priklausomybė nuo stebimo pseudo atsitiktinės sekos fragmento ilgio (sekos nuo 14 iki 18 bitų)

(3.12) paveiksluose atvaizduojami duomenys tik iki kol fragmento ilgis pasiekia 80 bitų, nes toliau didinant bitų skaičių gautiems rezultatams įtakos nebeturi. Fragmento ilgiui pasiekus tam tikrą bitų skaičių, aptikimo patikimumas tampa pastovus skaičius ir didinant fragmento ilgį, jis nebekinta. Iš gautų rezultatų matyti, kad maksimalus 100 % patikimumas pasiekiamas tik tada, kai nuskaityto fragmento bitų kiekis gerokai viršijus seką sugeneravusio generatoriaus pradinį bitų kiekį.

Siekiant įsitikinti atliktų eksperimentų duomenimis, buvo atliktas bandymas, siekiant nustatyti teisingai aptikto fragmento paklaidos pasiskirstymą. Atlikta dar 100 papildomų bandymų, (3.13) paveiksle pavaizduoti rezultatai. Ištininė linija simbolizuoja maksimalią gautą tikimybę eksperimento metu, punktyrinė linija simbolizuoja minimalią tikimybę eksperimento metu.



3.13 pav. Koreliacijos metodo vietos aptikimo tikimybės priklausomybė nuo stebimo pseudo sekos fragmento ilgio, pakartojus eksperimentą 100 kartų

Iš (3.13) paveikslo matyti, kad paklaidų pasiskirstymas yra ganėtinai nedidelis, svyruoja iki 10%, todėl (3.12) pateiktame paveiksle bandymų rezultatus galime laikyti teisingais.

3.5.3. Euklido atstumo algoritmas

Euklido atstumo algoritmas paremtas skirtumu tarp atraminės sekos ir ieškomo fragmento. Algoritmas realizuojamas nuskaitytą fragmentą atimant iš atraminės sekos ir skirtumo kvadratinės

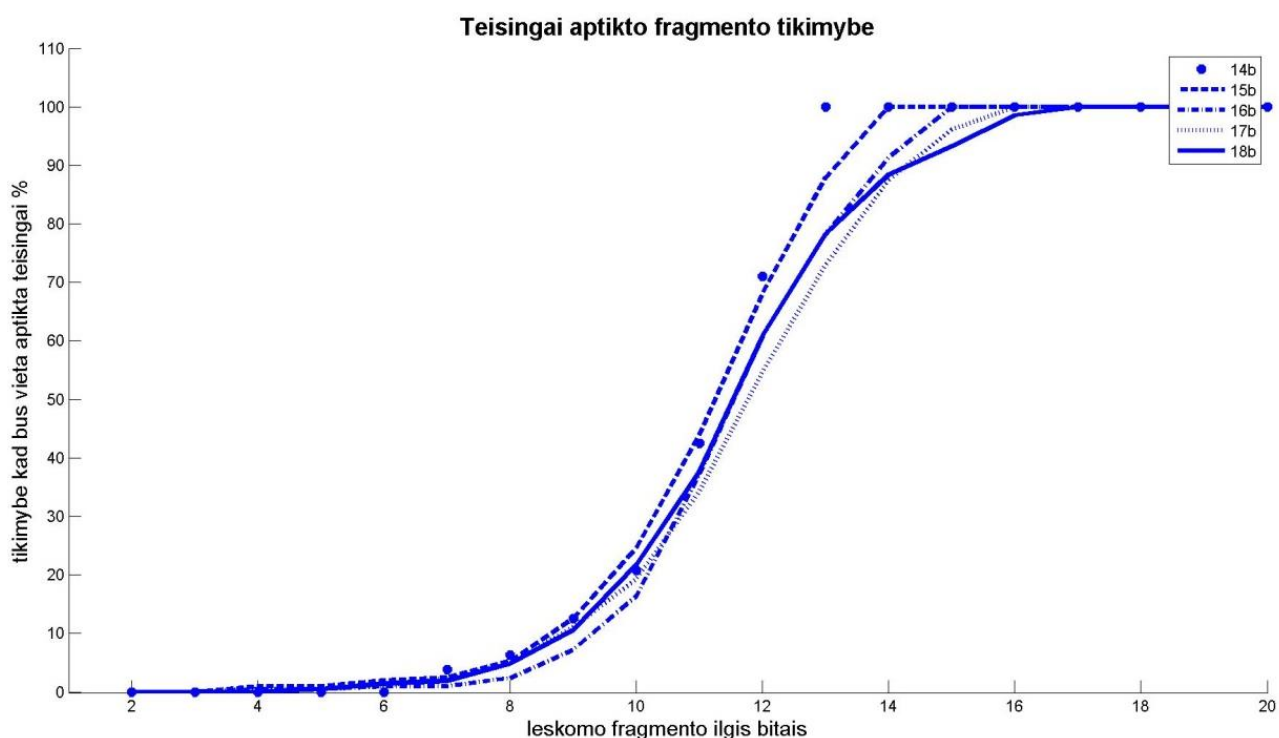
reikšmės susumuojamos. Toliau nuskaitytas fragmentas atraminės sekos atžvilgiu perstumiamas per vieną bitą ir veiksmas kartojamas iš naujo. Taip gaunamas masyvas, kurio mažiausia reikšmė atitinka fragmento poziciją sekoje. Algoritmą iliustruoja (7) formulė.

$$E(n) = \sum_{i=1}^{i+N} (x(n+i) - y(n+i))^2 \quad (7)$$

Šioje formulėje E atitinka euklido funkciją, N - tai ieškomo fragmento ilgis, x – atraminė seka, y – ieškomas fragmentas.

Kaip ir (3.4) skyriuje aptartame koreliacijos eksperimente, euklido atstumo algoritmas tikrinamas tokiu pačiu principu. Eksperimento metu imamos skirtingos sekos, naudojamų sekų duomenys pateikti (4) lentelėje. Fragmentai ieškomi tik iš pirmųjų 10 000 galimų fragmentų. Fragmento ilgis kinta nuo 2 bitų iki 500.

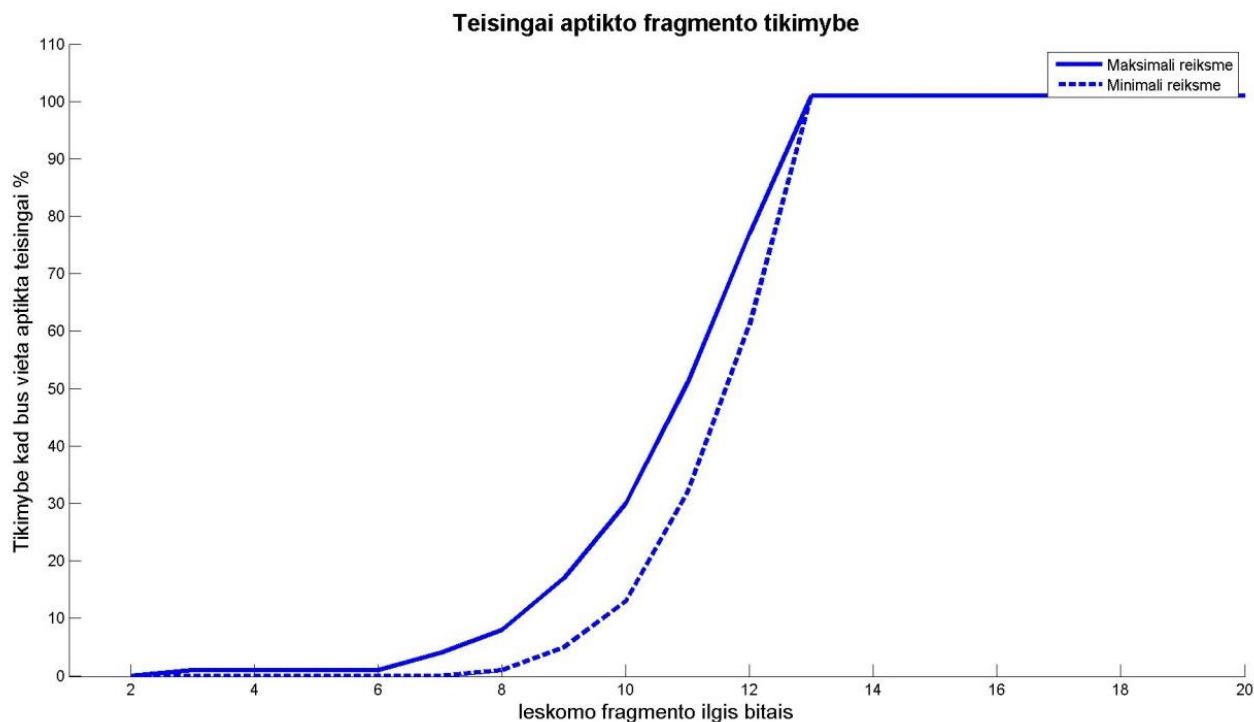
Šiuo atveju (3.14) paveiksluose atvaizduojama statistika iki 20 bitų fragmento ilgio, nes kaip ir koreliacijos atveju pasiekus tam tikrą fragmento ilgį patikimumo reikšmė nebekinta.



3.14 pav. Euklido atstumo metodo teisingo vietos nustatymo tikimybės priklausomybė nuo stebimo pseudo sekos fragmento ilgio (sekos nuo 12 iki 16 pagal Lentelę 4)

Iš gautų eksperimento rezultatų matyti, kad 100% patikimumo rezultatas pasiekiamas tada kai fragmento ilgis susivienodina su seką sugeneravusio generatoriaus bitų skaičiumi. Dėl šios

priežasties, reikalingo nuskaityti fragmento ilgis ganėtinai sutrumpėja, nes nereikalinga daug papildomų bitų. Šitokiu būdu mažinami skaičiavimo bei atminties resursai.

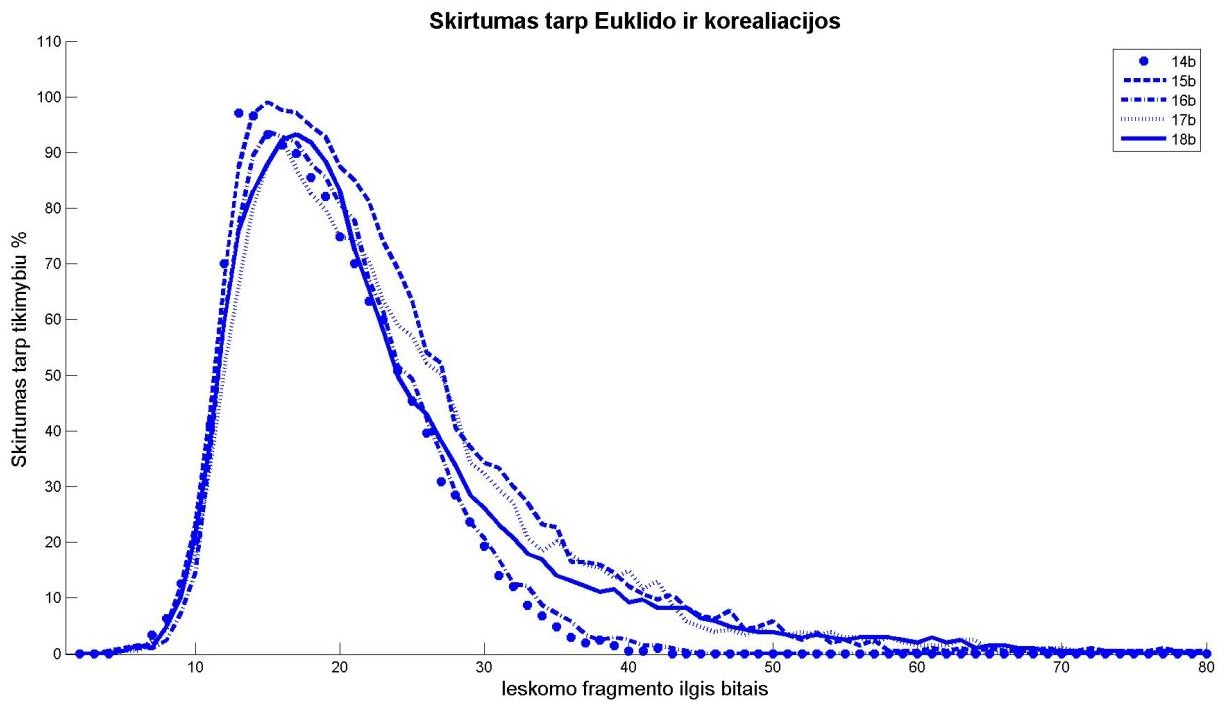


3.14 pav. Euklido metodo vietos aptikimo tikimybės priklausomybė nuo stebimo pseudo sekos fragmento ilgio, pakartojus eksperimentą 100 kartų

Kaip ir (3.13) paveiksle parodytų bandymų metu eksperimentas pakartojamas ir (3.14) paveiksle atvaizduojamas bandymo duomenų pasiskirstymas. Kaip galima pastebėti jis taip pat nėra didelis, iki 10%, todėl (3.13) paveiksle pavaizduoto eksperimento rezultatai yra teisingi.

3.5.4. Euklido ir koreliacijos algoritmų palyginimas

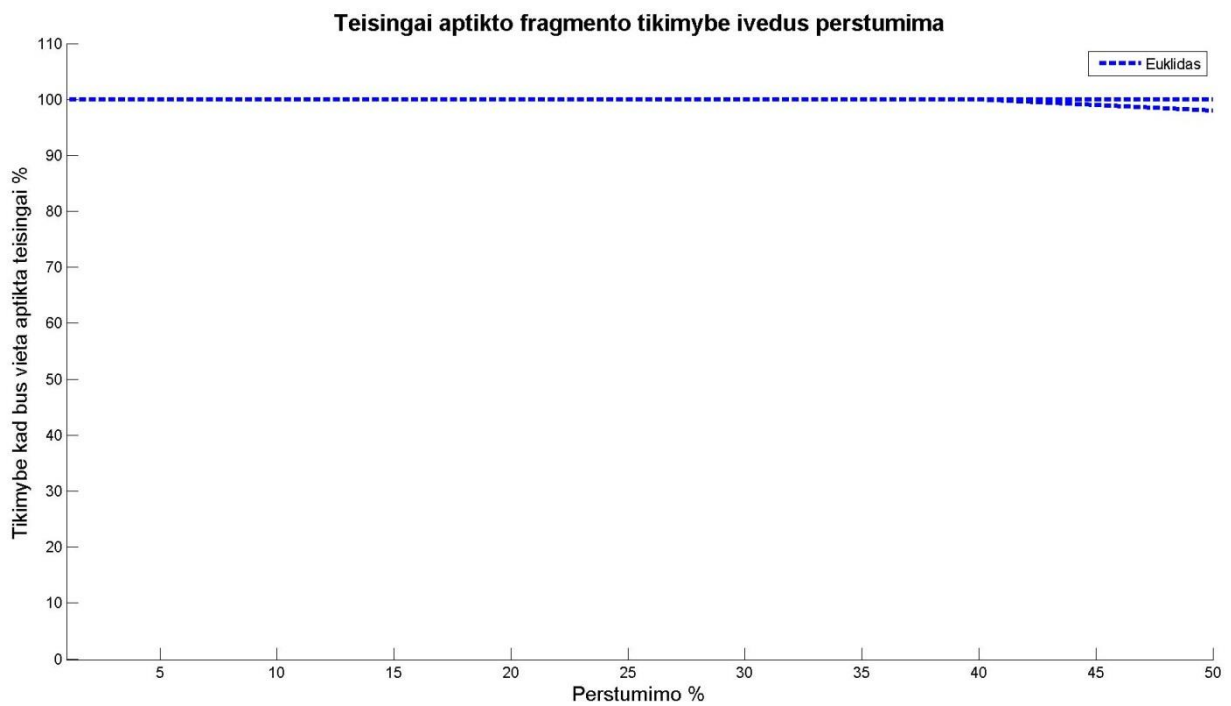
Šiame skyriuje bus toliau analizuojami koreliacijos ir euklido atstumo algoritmų skirtumai. (3.16) paveiksluose pateiktuose grafikuose pavaizduotas sekų patikimumo grafikų skirtumas tarp Euklido ir koreliacijos metodų. Grafikuose pateikta informacija, tai iš apskaičiuoto Euklido algoritmo patikimumo reikšmės atimama koreliacijos algoritmo apskaičiuotas patikimumas, tokiu būdu gaunamas patikimumo skirtumas.



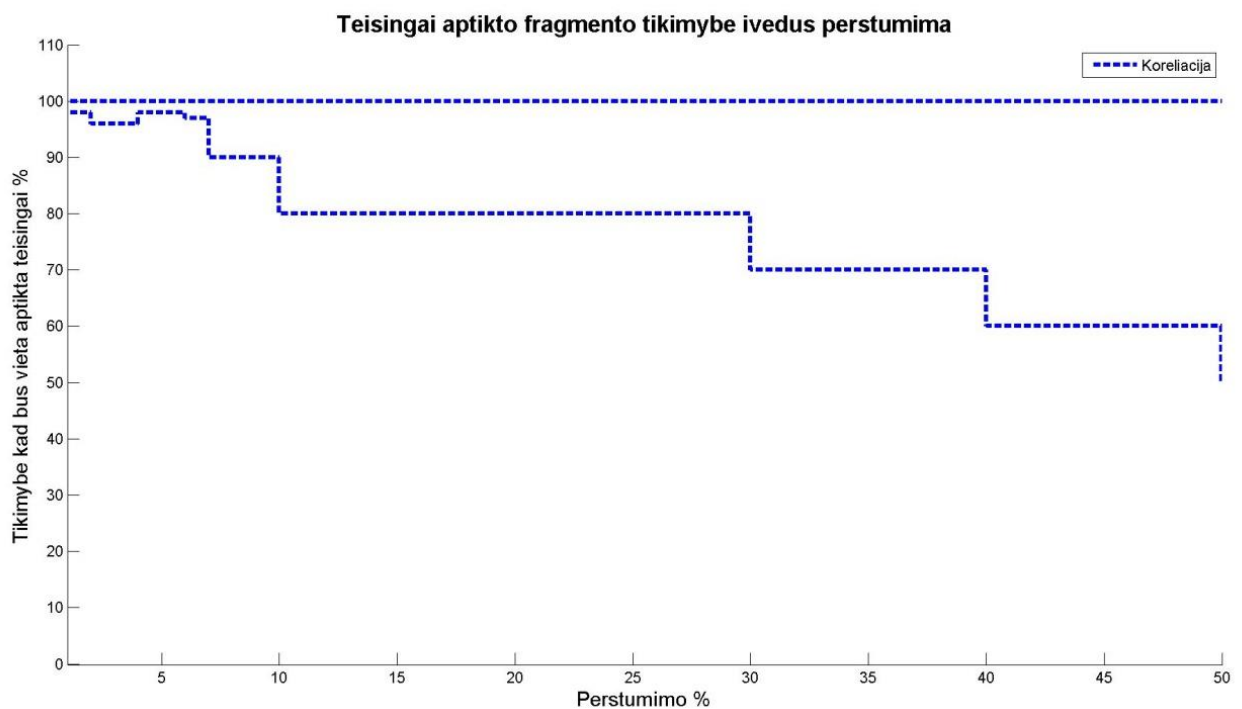
3.16 pav. Skirtumo tarp Euklido ir koreliacijos algoritmų teisingos vietos nustatymo tikimybių priklausomybė nuo stebimo fragmento ilgio, (sekos nuo 14 bitų iki 18 bitų)

Kaip galima pastebėti iš gautų rezultatų patikimumo skirtumas esant mažam bitų skaičiui gana didelis. Didžiajai daliai sekų, kai fragmento bitų skaičius neviršija arba pasiekia seką sugeneravusio generatoriaus bitų skaičių, patikimumo skirtumas siekia tarp 90 ir 100 procentų.

Toliau įvedama perstūmimo paklaida, tada kai jutiklio pozicija nesutampa su liniuotės. Iš praeitų eksperimentų matome, kad maksimalus galimas sekų patikimumas pasiekiamas jau ties 100 bitų riba. Siekiant, kad fragmento ilgis nė kiek neįtakotų eksperimento rezultatų, fragmento ilgis parenkamas 200 bitų. Sekos diskretizavimas parenkamas 100 išrankų bitui. Eksperimentas atliekamas duomenis parenkant iš keleto skirtingų pseudo atsitiktinių sekų. Bandymų rezultatai pateikti (3.17 ir 4.18) paveiksluose. Juose viršutinė grafiko riba simbolizuoja maksimalią galimą aptikimo tikimybę, apatinė - minimalią.



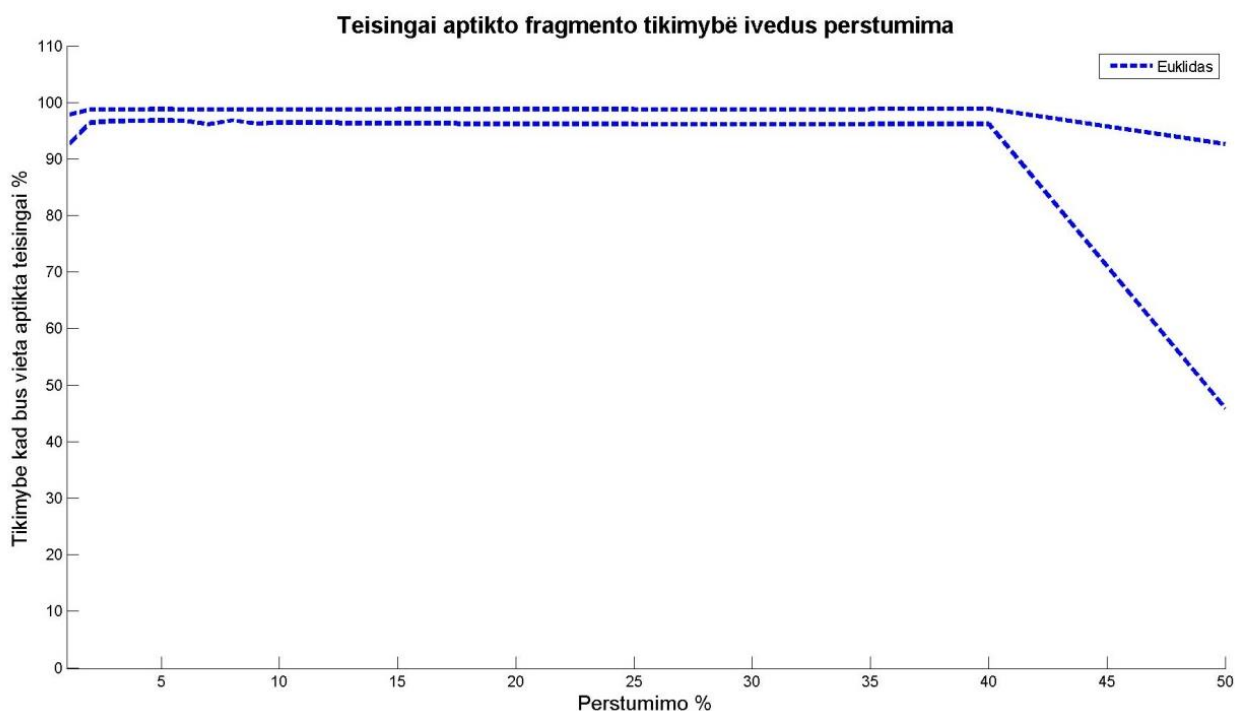
3.17 pav. Teisingos vietos aptikimo tikimybė nuo perstūmimo procento, naudojant Euklido algoritmą



3.18 pav. Teisingos vietos aptikimo tikimybė nuo perstūmimo procento, naudojant koreliacijos algoritmą

Kaip galima pastebėti iš (3.17) paveikslo, Euklido metodui sekos perstūmimas didelės įtakos neturi. Naudojant koreliacijos metodą, rezultatai gaunami daug prastesni. Kaip galima matyti iš

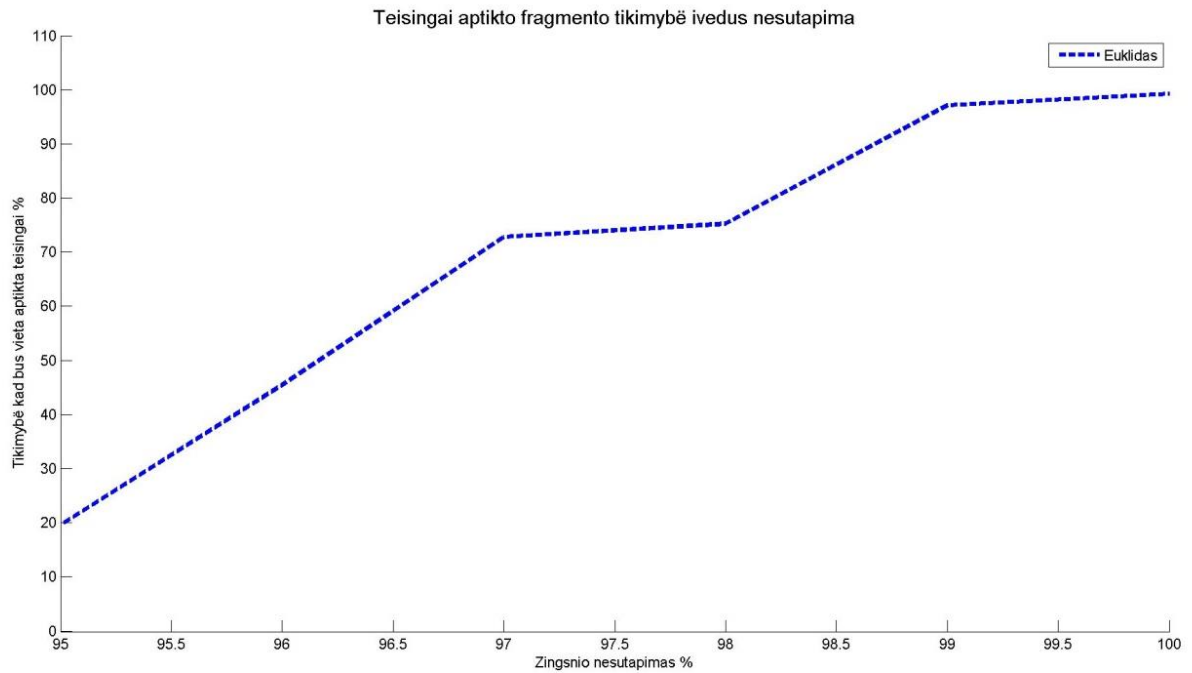
(3.18) paveikslo, teisingos padėties aptikimo tikimybės pasiskirstymas daug labiau pasiskirstęs, lyginant su Euklido algoritmu. Siekiant tikslumo eksperimentas pakartojamas, tačiau prie žemesnių fragmento ilgio reikšmių. Šiuo atveju eksperimentas atliekamas atsižvelgiant į (3.19) paveiksle pateiktas reikšmes ir eksperimentas atliekamas su viena 14 bitų seka, fragmento ilgio režiuose nuo 40 iki 50 bitų



3.19 pav. Teisingos vietos aptikimo tikimybė nuo perstūmimo procento, naudojant Euklido algoritmą, fragmentą sumažinus iki minimalios bitų reikšmės (40 – 50 bitų)

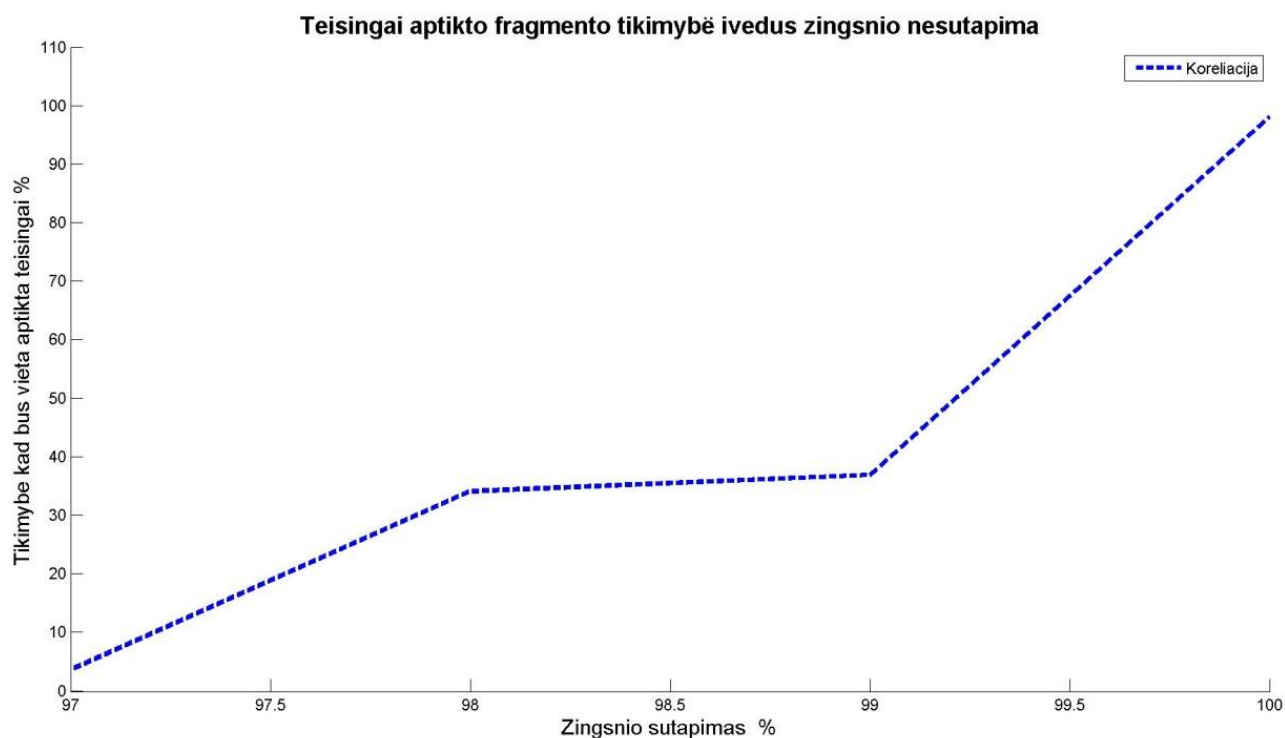
Kaip galima pastebėti iš (3.19) paveikslo, tik gana didelė perstūmimo paklaida, pasiekusi 50%, gana smarkiai įtakoja padėties aptikimą su Euklido atstumo algoritmu. Todėl galima teigti, kad Euklido atstumo algoritmas, pagal atliktus perstūmimo ir priklausomybės eksperimentus, yra pranašesnis nei koreliacijos algoritmas. Pastarasis, teisingam padėties aptikimui reikalauja iki 4 kartų daugiau papildomų bitų, siekiant aptikti teisingą padėtį. Atsparumas perstūmimo paklaidai taip pat daug mažesnis nei Euklido atstumo algoritmo, tai puikiai galima pastebėti iš (3.18) ir (3.19) paveikslų.

Atlikus sekos perstūmimo priklausomybės eksperimentą, reikalinga įvertinti kaip Euklido ir koreliacijos algoritmą įtakoja pikselio žingsnio nesutapimo paklaida. Eksperimentas atliekamas pasinaudojant prieš tai sukurtu modeliu ir sugeneruota 14 bitų pseudo atsitiktine seka. Eksperimentas atliekamas modeliuojant žingsnio netikslumą nuo 0 iki 5%, eksperimento rezultatai pateikti (3.20) ir (3.21) paveiksluose.



3.20 pav. Teisingos vietos aptikimo tikimybė nuo įvedus žingsnio nesutapimą, naudojant Euklido algoritmą

Kaip galima pastebėti iš (3.20) paveikslo, net menkiausia perstūmimo paklaida įtakoja Euklido atstumo algoritmo galimybę teisingai aptikti padėtį. Įvedus 2 procentų nesutapimo paklaidą, patikimumas krenta iki 70%. Jei nesutapimo paklaida dar didesnė, tai iki 5% patikimumas nukrenta iki ganėtinai žemos ribos, 20%.



3.21 pav. Teisingos vietos aptikimo tikimybė nuo įvedus žingsnio nesutapimą, naudojant koreliacijos algoritmą

Atlikus analogišką eksperimentą su koreliacijos algoritmu, (3.21) paveiksle pateikiami jo rezultatai. Juose galime pastebėti dar didesnius patikimumo sumažėjimus. Jau esant 1%, patikimumas sumažėja iki 35%, kai tuo tarpu Euklido algoritmo atveju prie 1% paklaidos patikimumas buvo 95%.

Atsižvelgiant į visus atliktus eksperimentus, jutiklio ilgio tyrimo, perstūmimo paklaidos, žingsnio nesutapimo paklaidos, matome, kad Euklido atstumo algoritmas yra gerokai pranašesnis už koreliacijos algoritmą, patikimumo atžvilgiu. Euklido atstumo algoritmas reikalauja mažiau papildomų duomenų padėčiai nustatyti, perstūmimo paklaida Euklido atstumo algoritmui beveik neturi jokios įtakos. Jutiklio žingsnio nesutapimo paklaidos įtaka, lyginant abudu algoritmus, Euklido metodo atveju yra daugiau nei du kartus geresnė.

3.5.5. Inversinio LFSR (Linear Feedback Shift Register) generatoriaus algoritmas

Šioje darbo dalyje tiriamas padėties nustatymas pasinaudojus LFSR registru ir nuskaityto kodo reikšmę. Tam naudojamas toks pats LFSR generatorius koks buvo panaudotas pradinei sekai sugeneruoti, arba jo atvirkštinė versija. Atvirkštinio LFSR generatoriaus polinomialai perskaičiuojami pagal (8) formulę [18].

$$\{n n - C n - B n - A 0\} \quad (8)$$

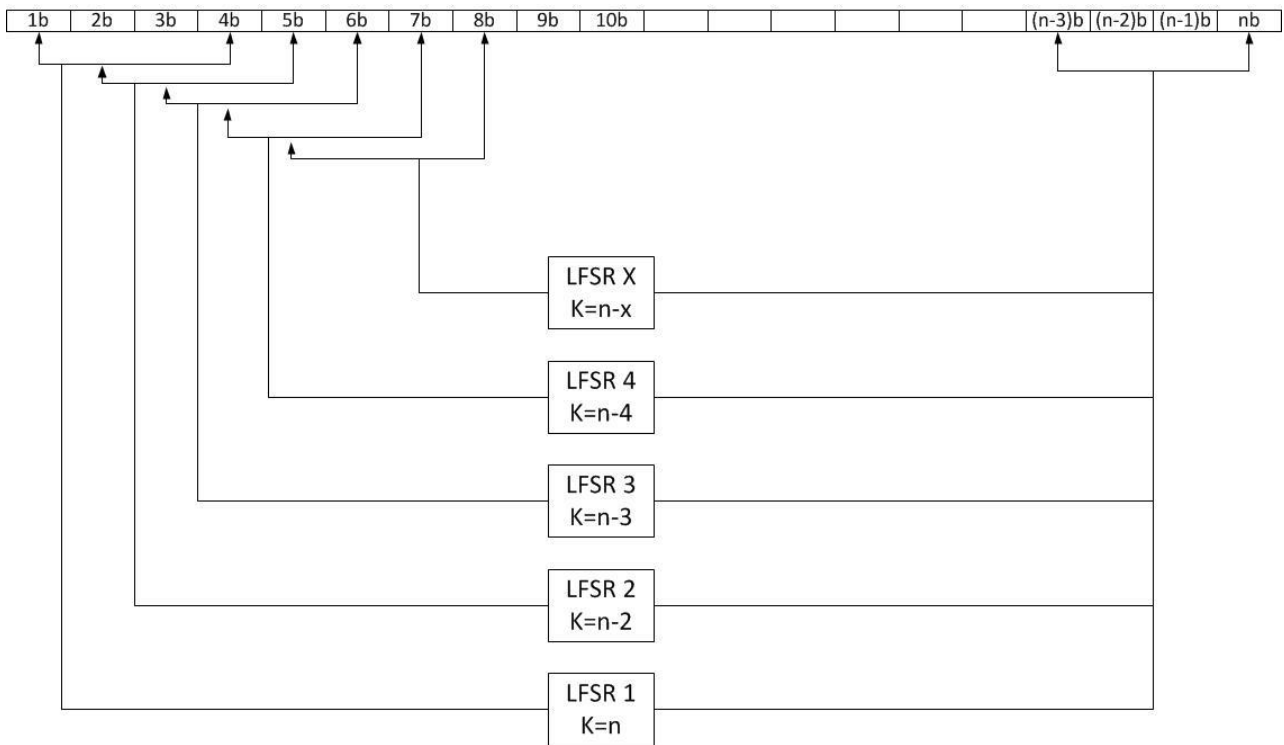
Čia n - tai generatoriaus bitų skaičius, A , B ir C - tai pirminio generatoriaus polinomialai. Perskaičiuotas atvirkštinis generatorius nuo pradinio skiriasi tuo, kad jo generuojama seka yra identiška pradiniam generatoriui, tačiau priešingai nei pirminė, generuojama į priešingą pusę.

Toliau Matlab programinės įrangos pagalba sukuriama programa, kurios pagalba LFSR generatoriumi pagrįsta nuskaityto kodo paieškos sistema. Pradinė seka naudojama tokia pati kaip ir koreliacijos atveju, eksperimentas atliekamas 100 kartų. Kiekvieno eksperimento metu, algoritmo gauta vieta aptinkama tiksliai ir be klaidų, taip pat fiksuojamas paieškai reikalingas laikas. Laiko fiksavime buvo gana didelis svyravimas, tačiau vidutinė skaičiavimo trukmė gauta apie 3,9s. Didelį laiko skaičiavimo svyravimą galima paaiškinti tuo, kad algoritmas paremtas skaičiavimo stabdymu, aptikus atitinkamą padėtį. Nuskaitomai padėčiai esant sekos pradžioje skaičiavimų laikas būna daug trumpesnis, nei ieškomai padėčiai esant sekos pabaigoje. Didelės vidutinės duomenų apdorojimo trukmės priežastis galėjo būti dėl to, kad Matlab programinei įrangai parašytas kodas visiškai neoptimizuotas skaičiavimų greičio atžvilgiu. Funkcijos buvo parašytos siekiant įgyvendinti paieškos funkcionalumą, kad būtų galima įsitikinti algoritmo veikimu, taip pat atlikti eksperimentus, skirtus klaidų įvertinimui. Perkėlus algoritmą į aparatinę įrangą, bei optimizuojant veikimą greičio atžvilgiu, padėties aptikimo laikas turėtų gerokai sumažėti.

Toliau atliekant eksperimentus buvo tikrinamas algoritmo atsparumas klaidoms. Pirmiausia buvo atliktas bandymas nuo pavienių klaidų. Įvedus bent vieną pavienę klaidą, algoritmas nebesugebėdavo aptikti teisingos padėties. Dėl šios priežasties algoritmas visiškai nėra atsparus klaidoms. Siekiant išspręsti patikimumo problemą, LFSR algoritmas yra patobulinimas, plačiau apie tai aprašyta (3.5.6) skyriuje.

3.5.6. Patobulintas inversinio LFSR generatoriaus algoritmas

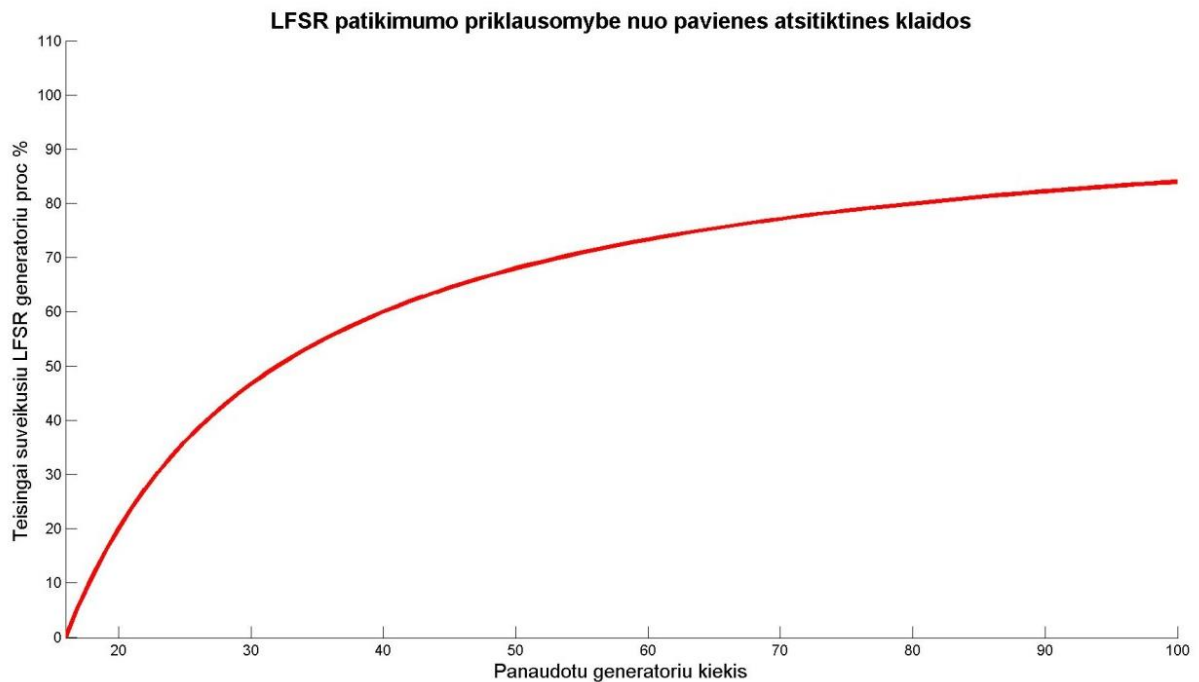
Mažo atsparumo klaidoms problema sprendžiama taip pat, kaip ir koreliacijos atveju, didinant nuskaityto kodo bitų skaičių ir naudojant daugiau generatorių, kurie dirbtų lygiagrečiai. Blokinė tokio algoritmo realizavimo schema pavaizduota (3.22) paveiksle.



3.22 pav. LFSR paieškos algoritmo principinė schema

Kadangi nuskaitytas kodas yra daug ilgesnis už generatoriaus bitų skaičių ir į kiekvieną generatorių perduodama skirtinga nuskaityto kodo dalis. Nuskaitytame kode įsivėlus klaidai rezultate tik dalis turimų generatorių parodys neteisingą padėtį. Todėl galima teikti, jeigu didžioji dalis naudojamų generatorių parodymai sutampa, vadinasi tai yra ieškoma kodo padėtis.

Siekiant įsitikinti algoritmo patikimumą pavienėms klaidoms, atliekamas eksperimentas. Eksperimento metu tiriama kaip nuskaityto kodo ilgis įtakoja patikimumo procentą, esant vienai pavienei klaidai. Eksperimento rezultatas pateiktas (3.23) paveiksle.

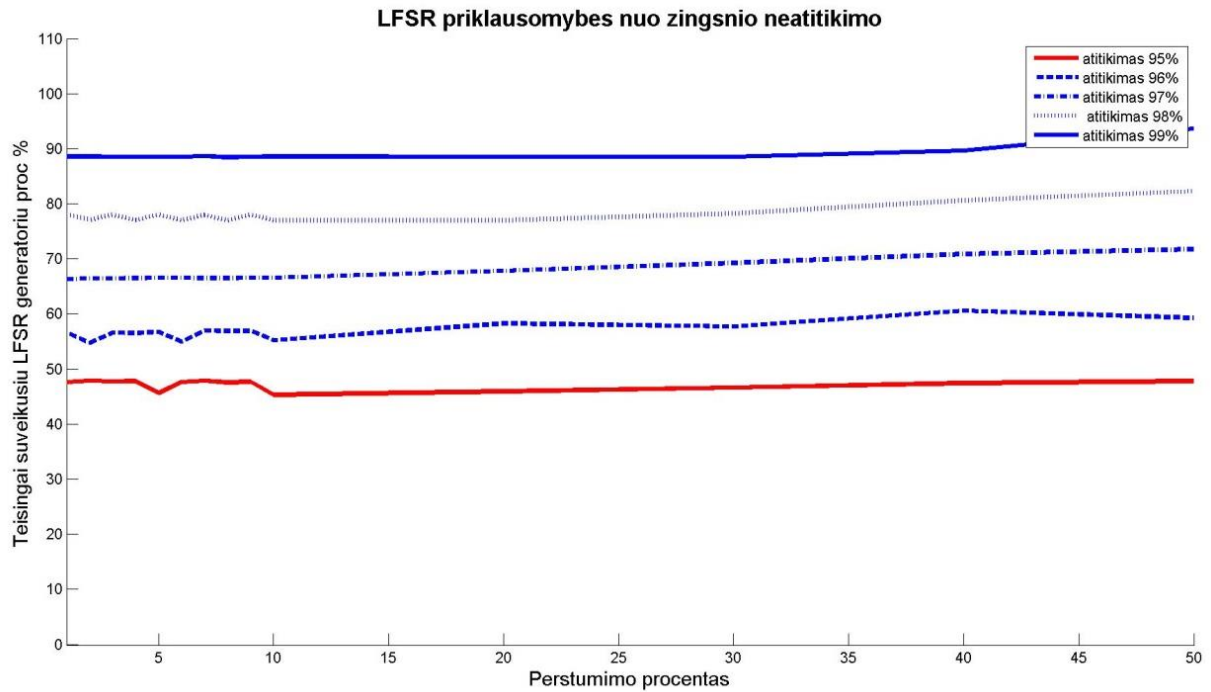


3.23 pav. LFSR generatoriaus tikslumo priklausomybė nuo pavienės paklaidos

Kaip galime pastebėti iš (3.23) teisingas padėties aptikimas priklauso nuo fragmento ilgio ir tuo pačiu, nuo naudojamų papildomų generatorių skaičiaus. Naudojant 40 bitų fragmentą, teisingai suveikusių generatorių skaičius, esant vienai klaidai nuskaitytame kode, pakylą aukščiau 50% ribos. Esant šitokiam santykiui, jau galima atlikti tikslų padėties nustatymą. Didinant papildomų bitų skaičių, šitas procentas dar labiau didėja. Pasiėkus 100 bitų fragmento ilgį, teisingai suveikusių generatorių procentas priartėja iki 90%, taip pat matome, kad šioje vietoje kreivė vis labiau pradeda tiesėti, todėl norint gauti dar didesnę atsparumą klaidoms, reikėtų didinti fragmento ilgį neproporcingai daug.

Klaidingu bitu buvo naudojamas 16 bitas, 16 bitų sekoje. Šitokia klaidingo bito pozicija buvo pasirinkta tam, kad būtų įtakota kuo daugiau generatorių. Iš gautų rezultatų matome, kad norint aptikimo algoritmą padaryti atsparų bent vienam neteisingam bitui, reikalingas nuskaitytų bitų skaičius turi padidėti daugiau nei du kartus.

Toliau siekiama įsitikinti, kaip šis algoritmas veikia ir yra įtakojamas perstūmimo ir žingsnio neatitikimo paklaidos. Eksperimentas atliekamas su 100 bitų fragmentu, naudojant (3.22) paveiksle pavaizduotą paieškos principą. (3.24) paveiksle pavaizduoti eksperimento rezultatai. (3.24) paveiksle taip pat įvertinta ir perstūmimo paklaida, bet kaip galima pastebėti iš kreivių, ji rezultatams įtakos jokios neturi. Vienintelis rezultatus įtakojantis veiksnys, tai žingsnio neatitikimo paklaida, kuri su kiekvienu neteisingu procentu, padėties aptikimo tikslumą įtakoja beveik dešimčia procentų.

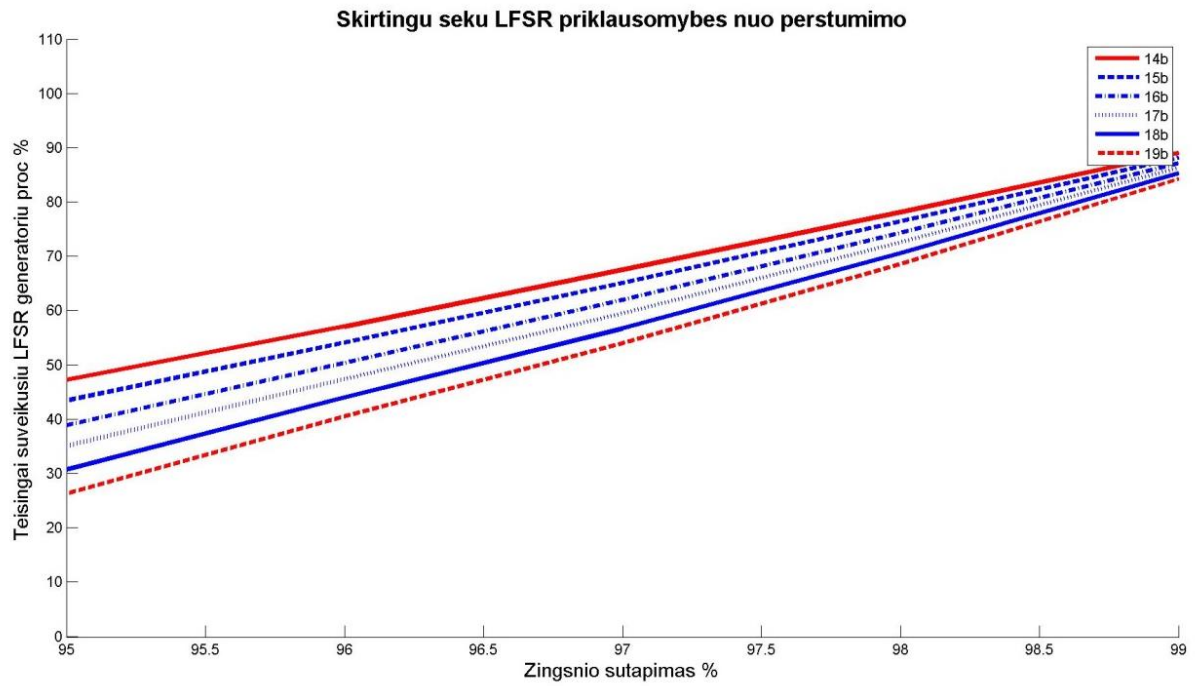


3.24 pav. Teisingai aptikusiu padėtį generatorių priklausomybė nuo žingsnio netikslumo

(3.24) paveiksle pavaizduotas tikslumas galimas tik patobulinus (3.22) paveiksle pavaizduotą sistemą. Kaip kad buvo minėta (3.4.2) skyriuje, žingsnio nesutapimo paklaida galutiniame rezultate įtakoja pavienio neteisingo bito atsiradimą, po kurio visa seka detektuojama kaip persislinkimas per vieną poziciją. Priklausomai nuo nesutapimo procento ir fragmento ilgio, šitie nesutapimai viename fragmente gali kartotis keletą kartų, tačiau tolimesniems rezultatams galioja dėsningumas, kuris leidžia dirbtinai patikslinti padėtį, nors jutiklis ir rodo neteisingai. Po kiekvieno sekos virsmo, sekantys generatoriai poziciją nustatys su vienos išrankos paklaida, jei algoritme įvesime sąlygą, kad keletas, vienas po kito einančių generatorių, nustato poziciją su vienoda paklaida, nei prieš tai sekoje buvę generatoriai, taip pat jei ta paklaida nėra didesnė nei viena išranka, galima traktuoti, kad pozicija aptikta teisingai ir atsiradusi paklaida yra eliminuojama dirbtinai.

Šis algoritmo tobulinimas išsprendžia atsiradusią patikimumo problemą, tačiau dėl reikalingo papildomo loginio įtaiso paklaidos tikrinimui, išauga sistemos reikalavimai skaičiavimams taip pat gali sumažėti greitaveika. Šitoks paklaidos tikrinimas labiausiai gali atsiliiepti sistemoms, pagrįstoms mikrovaldikliais, dėl jų ribotų galimybių užduotis vykdyti lygiagrečiai. Šios problemos sprendimas būtų - paieškos algoritmo realizavimas su programuojamu loginiu įrenginiu. Jame galima būtų nesunkiai realizuoti didelį kiekį atskirų LFSR registru, kurie priešingai nei procesoriuje, galėtų dirbti lygiagrečiai ir nepriklausomai vienas nuo kito. Šitokiu būdu galima sumažinti ne tik bendrą algoritmo veikimo laiką, bet ir konkretaus LFSR registro veikimo laiką.

Prieš tai aptarti bandymai dar buvo atlikti su įvairaus ilgio pseudo atsitiktinėmis sekomis. Panaudotos sekos siekė nuo 14 bitų iki 19 bitų. Gauti bandymo rezultatai pateikti (3.25) paveiksle.



3.25 pav. Skirtingų sekų priklausomybė nuo žingsnio neatitikimo

4. LINIJINIO ABSOLIUTINIO POSLINKIO ALGORITMŲ REALIZAVIMAS

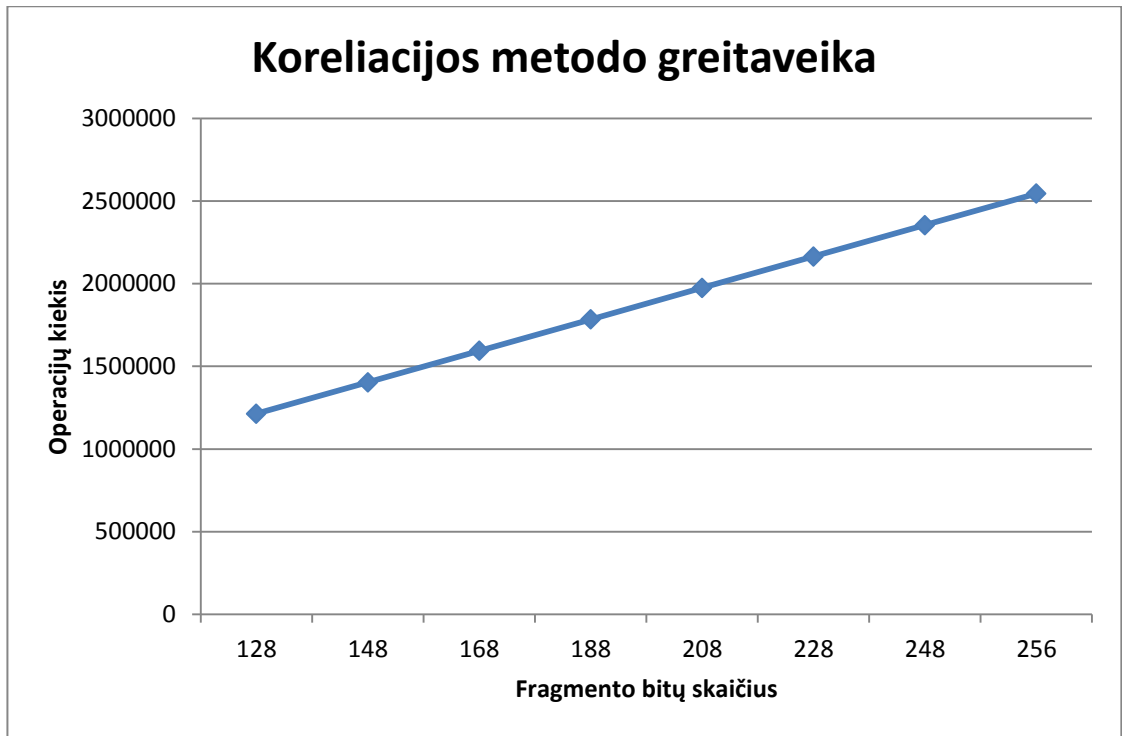
4.1. Galimos aparatinės įrangos alternatyvos

Toliau atliekami eksperimentai ir greitaveikos simuliacijos su realia aparatine įranga. Koreliacijos, euklido atstumo ir LFSR generatorių padėties aptikimo algoritmai pritaikomi veikti su mikrovaldikliais ir programuojamais loginiais įrenginiais. Greitaveikai su mikrovaldikliu parenkamas arm šeimos STM32F407 mikrovaldiklis turintis darbinį dažnį iki 168MHz. Šis mikrovaldiklis parenkamas, dėl pakankamai didelės greitaveikos taip pat dėl to, kad savyje turi DSP posistemę, kuri galėtų būti išnaudojama greitaveikos didinimui. Eksperimento tikslas - išsiaiškinti algoritmų tarpusavio greitaveiką mikrovaldiklių paremtoje sistemoje. Taip pat, palyginti rezultatus su programuojamos logikos įrenginiais. Programuojamos logikos įrenginys algoritmų realizavimui parenkamas alteros cyclons II loginis įtaisas.

4.2. Mikroprocesorinėmis priemonėmis

Padėties aptikimas naudojant pseudo atsitiktinę seką realizuojamas pasitelkiant mikrovaldiklius. Šiame skyriuje bus atlikti eksperimentai, kurių metu bus lyginama greitaveika tarp trijų skirtingų algoritmų: koreliacija paremto, Euklido atstumu, ir LFSR generatoriais. Eksperimentams atlikti naudojama ARM tipo ST firmos mikrovaldikliai.

Pirmiausia atliekamas eksperimentas, kai paieškos algoritmas paremtas Koreliacija. Eksperimento metu parenkami pradiniai duomenys, tai - 120 bitų ilgio fragmentas iš 2000 bitų ilgio pseudo atsitiktinės sekos. Šis pradinis eksperimento scenarijus bus naudojamas ir tolimesniuose eksperimentuose, siekiant objektyviai išsiaiškinti, kuris iš tiriamų algoritmų yra greičiausiai vykdomas. Palyginimas atliekamas palyginus operacijų skaičių, reikalingą įvykdyti pasirinktam algoritmui. Atliekant eksperimentą pirmiausia atliekamas fragmento bitų ilgio didinimas, o sekos ilgis nekeičiamas, eksperimento rezultatai pateikiami (4.1) paveiksle.



4.1 pav. Koreliacijos metodo greitaveikos priklausomybė nuo fragmento ilgio.

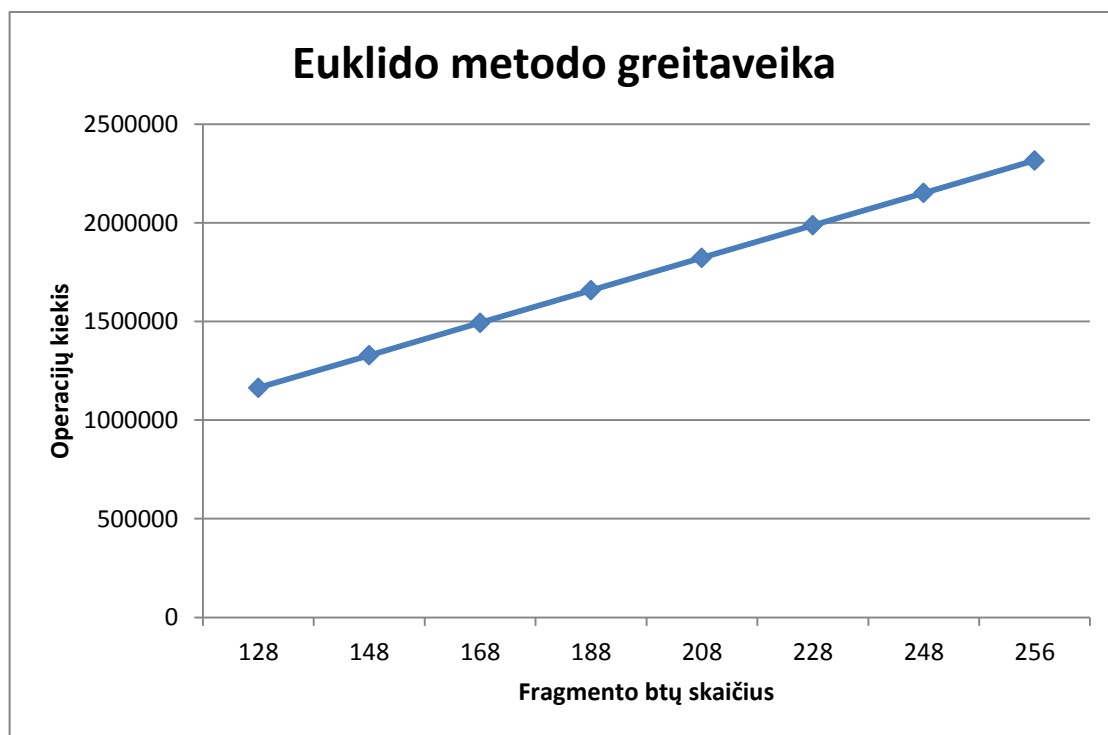
Kaip galima pastebėti iš (4.1) paveikslo, padidinus koreliacijai naudojamo fragmento ilgį dvigubai, skaičiavimo trukmė taip pat padvigubėja. Toliau atliekamas bandymas su kintančių sekos ilgiu. Bandymo rezultatai pateikti (4.2) paveiksle.



4.2 pav. Koreliacijos metodo greitaveikos priklausomybė nuo sekos ilgio

Kaip ir (4.1) paveiksle pavaizduoto bandymo metu, padvigubinus sekos ilgį skaičiavimo greitaveika mažėja dvigubai. Dėl (4.1) ir (4.2) paveiksluose pavaizduotų rezultatų galima teigti, kad naudojamas sekos fragmentų ir sekos ilgių reikšmės tiesiškai įtakoja algoritmo greitaveiką.

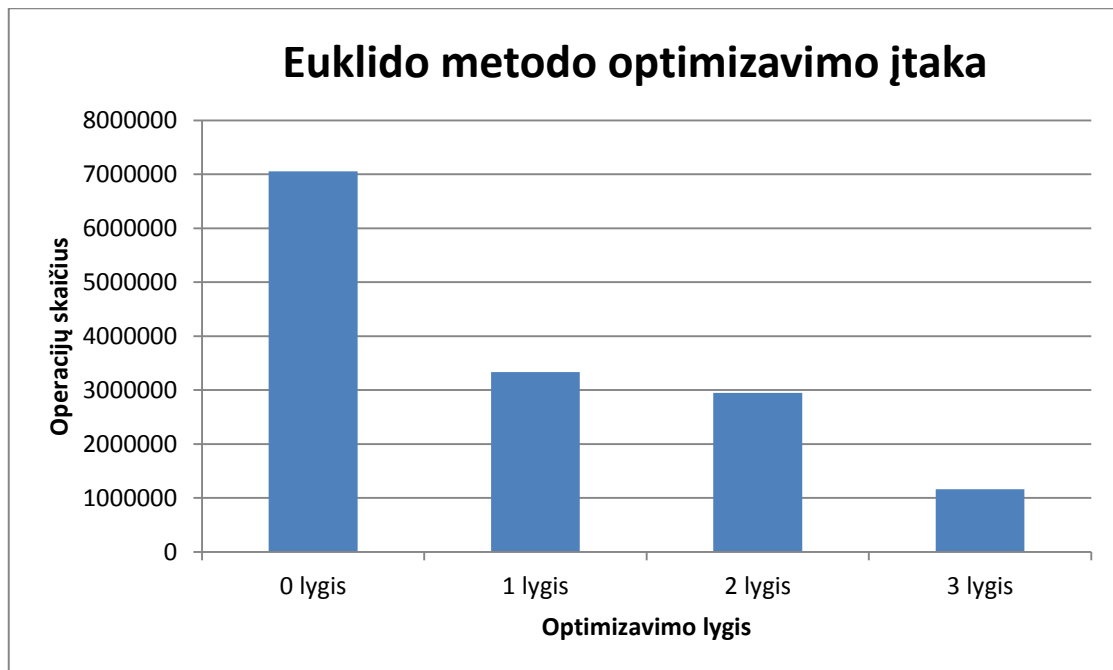
Tolimesnėje darbo dalyje atliekamas analogiškas eksperimentas su Euklido atstumu paremtu padėties aptikimo algoritmu. Kaip ir prieš tai atliktame bandyme parenkamas fragmento ilgis lygus 128 bitams, sekos ilgis, kuriame ieškomas fragmentas, taip pat lygus 2000 bitų. Pirmiausia atliekamas bandymas esant pastoviai pseudo atsitiktinei sekai, bandymo rezultatai pateikti (4.3) paveiksle.



4.3 pav. Euklido metodo greitaveikos priklausomybė nuo fragmento dydžio

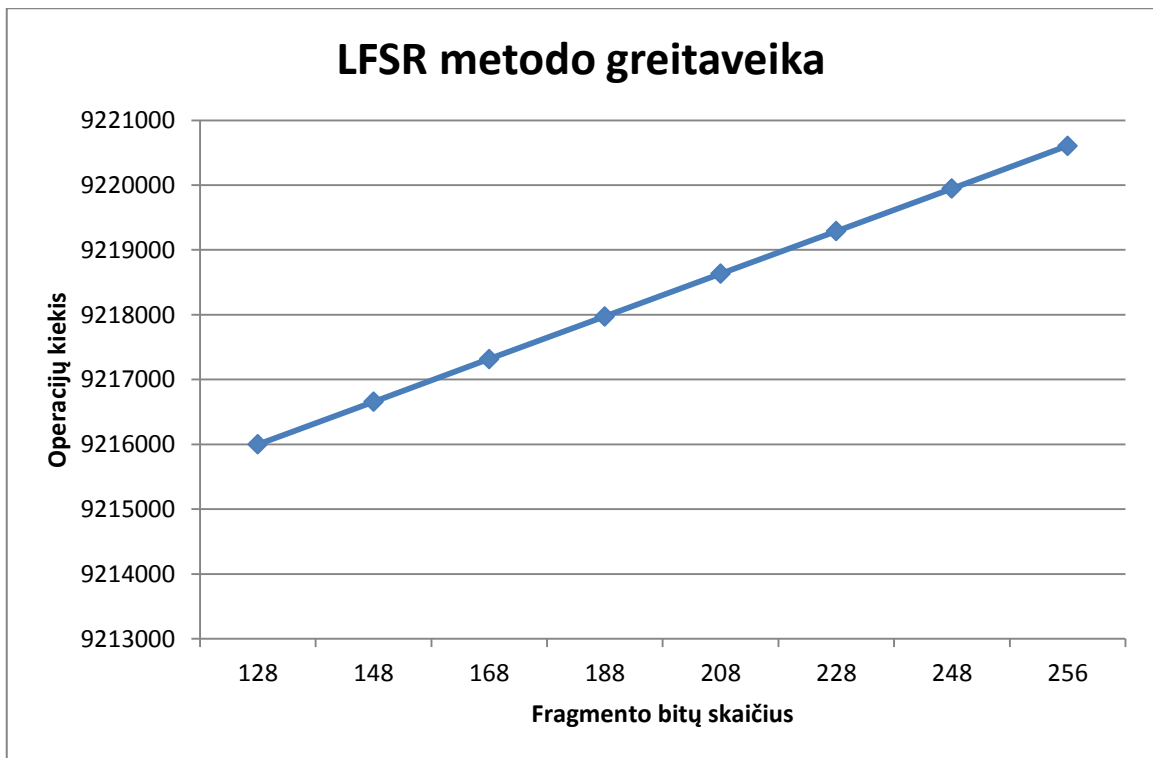
Iš pateikto (4.3) paveikslo, galime matyti, kad ir šiuo atveju, greitaveikos priklausomybė nuo fragmento ilgio yra tiesinė. Atlikus eksperimentą, kai fragmento ilgis lieka nekintantis, bet pačios sekos ilgis kinta, pasikartoja ta pati tendencija, kaip ir prieš tai atliktuose bandymuose. Sekos ilgio priklausomybė yra tiesiogiai proporcinga algoritmo greitaveikai. Sekai padvigubėjus algoritmo veikimo greitaveika sumažėja du kartus.

Koreliacija paremto algoritmo taip pat ir Euklido atstumu paremto algoritmo programos kodai buvo išbandyti prie skirtingų mašininio kodo kompiliavimo optimizavimo lygių. Koreliacijos algoritmo greitaveikai, kompiliatoriaus optimizavimo lygiai jokios įtakos neturėjo. Šitoks bandymo rezultatas įtakoja tai, kad koreliacinei funkcijai skaičiuoti naudojama maksimaliai optimizuota ARM sistemos bibliotekos funkcija. Euklido atstumo algoritmui įgyvendinti nebuvo naudojamos optimizuotos bei specializuotos funkcijos, dėl šios priežasties kompiliatoriaus optimizavimo įtaka yra akivaizdi. Grafiškai tai pateikta (4.4) paveiksle.



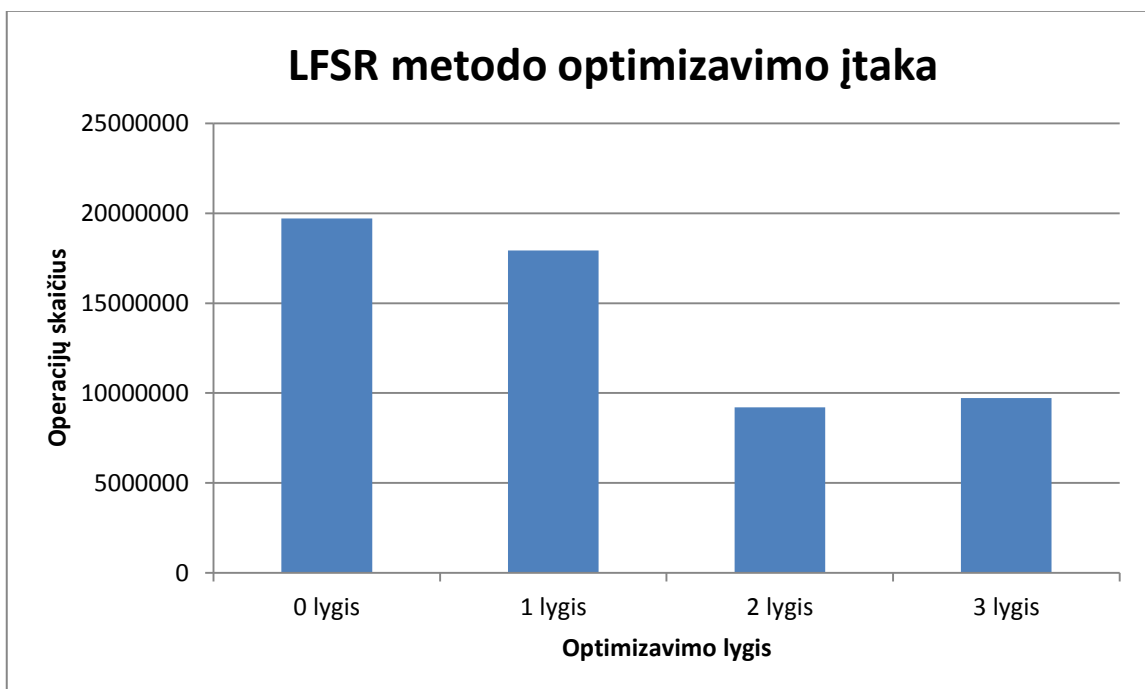
4.4 pav. Euklido metodo optimizavimas

Galiausiai atliekamas bandymas su LFSR generatoriumi paremtu algoritmu. Prieš tai aptarti koreliacijos ir Euklido algoritmai turėjo vieną panašumą - abiejų greیتaveika nepriklausė nuo ieškamos pozicijos pseudo atsitiktinėje sekoje. Jei nuskaitytas fragmentas būtų paimtas iš sekos pradžios arba iš galo, paieškos trukmei tai jokios įtakos neturėtų. LFSR generatoriais paremto algoritmo veikimas priešingas, jei nuskaitytas fragmentas yra sekos pradžioje, tai padėtis aptinkama greičiau, lyginant su tuo, jei nuskaitytas fragmentas būtų sekos pabaigoje. Atsižvelgus į ką tik paminėtą faktą, algoritmo greیتaveika fiksuojama esant didžiausiai trukmei. Tai reiškia, kad nuskaitytas fragmentas bus imamas iš pačios sekos pabaigos, todėl LFSR generatorius norėdamas rasti nuskaityto fragmento poziciją turės paėiliui sugeneruoti visą naudojamą liniuotėje seką. Šio bandymo metu taip pat panaudotas fragmento ilgis nuo 128 bitų bei sekos ilgis 2000, pirmiausia sekos ilgis laikomas pastoviu dydžiu, kinta tik fragmento dydis, po to atliekamas bandymas ir su kintančiu sekos ilgiu. Bandymo rezultatai pateikti (4.5) paveiksle.



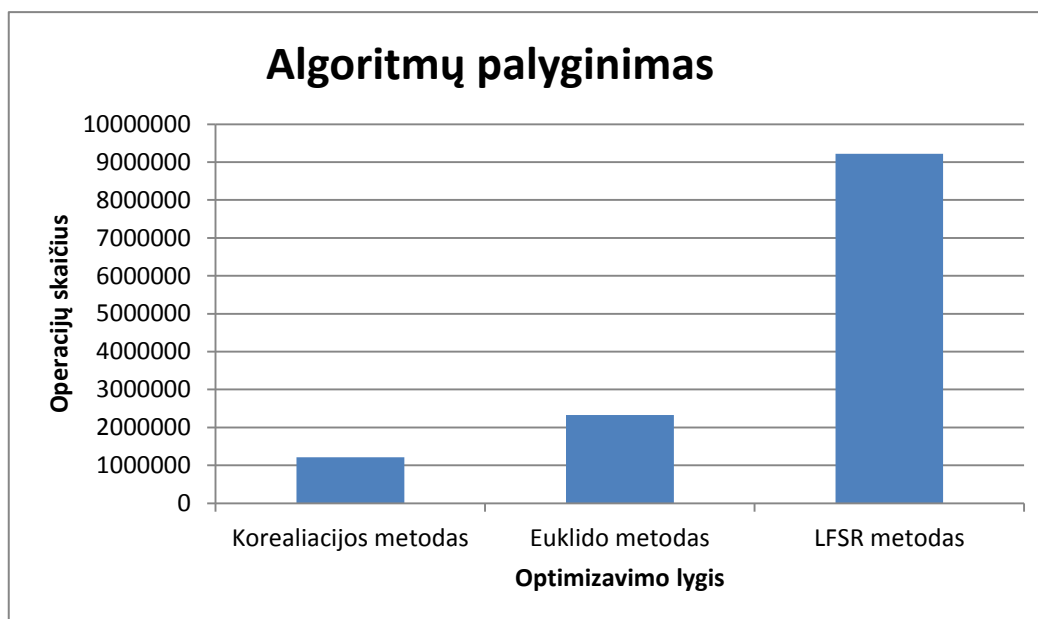
4.5 pav. LFSR generatoriais paremto algoritmo greitaveikos tyrimas.

Kaip galime pastebėti iš pateikto (4.5) paveikslo, kartojasi tokia pati tendencija kaip ir prieš tai atliktuose algoritmuose - greitaveikos dydis tiesiogiai proporcingas fragmento dydžiui, padvigubinus fragmento dydį - greitaveika sumažėja du kartus. Bandymas buvo pakartotas su skirtingais optimizavimo lygiais, bandymo rezultatai pateikti (4.6) paveiksle.



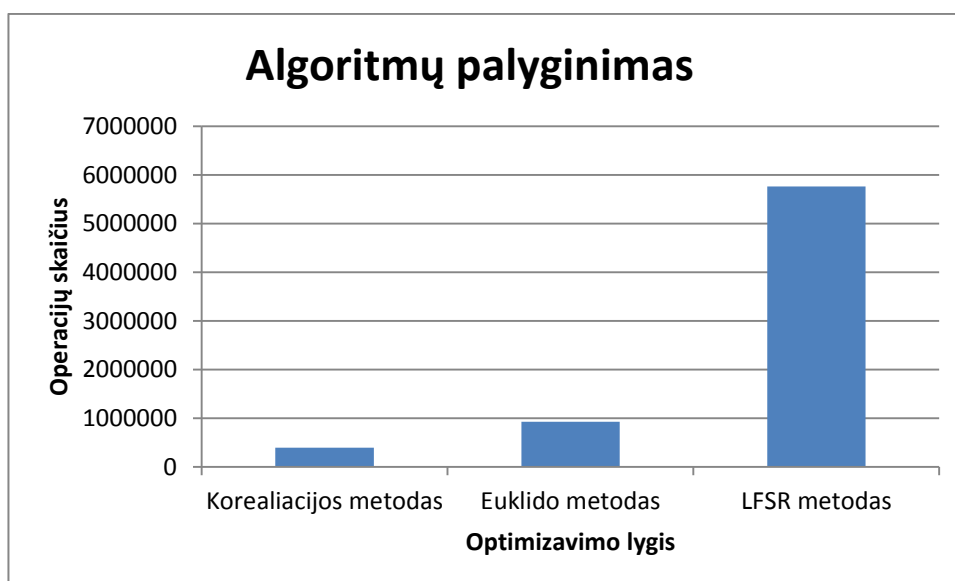
5.6 pav. Optimizavimo įtaka LFSR metodo greitaveikai

Nors visiems trimis algoritmams galioja ta pati tendencija, tiesioginė greitaveikos priklausomybė nuo nuskaitytų duomenų kiekio, tačiau greitaveika prie vienodų parametrų skiriasi. (4.7) paveiksle galime pastebėti greitaveikos skirtumus tarp visų trijų algoritmų.



5.7 pav. Greitaveikos skirtumai tarp trijų algoritmų.

Kaip galime pastebėti iš (4.7) paveikslo, koreliacijos algoritmas yra daug greitesnis nei likę du, tačiau šios eksperimento sąlygos yra lygios, nėra atsižvelgiama į optimalų algoritmų veikimo diapazoną. Kadangi Euklido atstumu paremtam algoritmui reikalingas daug mažesnis bitų kiekis fragmento ilgis sumažinamas iki 20 bitų. Analogiškai pasielgiama ir su koreliacijos bei LFSR algoritmais, pastarojo bitų kiekis pakeičiamas į 80 bitų, koreliacijos į 80 bitų. Atliekamas greitaveikos vertinimas, eksperimento rezultatai pateikti (4.8) paveiksle.



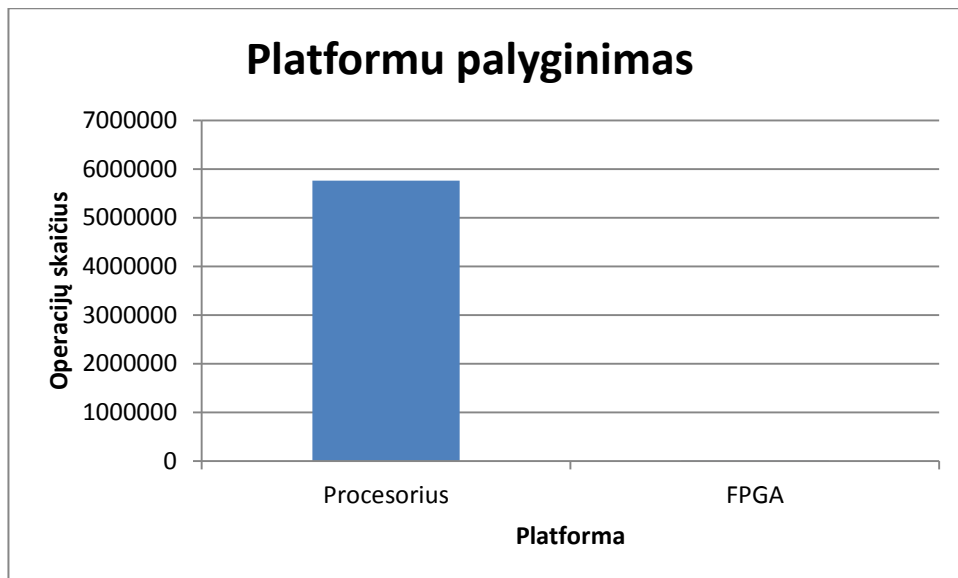
4.8 pav. Algoritmų palyginimas parinkus optimalius parametrus

Kaip matome iš pateikto paveikslo, parinkti optimalūs parametrai greitaveiką įtakoją daugiau, bendros tendencijos nekeičia. Koreliacijos algoritmas vis tiek lieka pranašesnis už likusius du. LFSR algoritmas net keletą kartų atsilieka nuo savo konkurentų. Šis atsilikimas įtakojamas fakto, kad didinant papildomų bitų skaičių, kiekvieną lygiagreto generatoriaus paiešką reikia atlikti paeiliui.

4.3. Programuojamos logikos priemonėmis

Šiame skyriuje aptariama kaip tiriamus padėties aptikimo algoritmus būtų galima realizuoti loginių įtaisų pagalba. Loginiai įtaisai, skirtingai nei mikrovaldikliai, turi du pranašumus. Pirmasis pranašumas, tai galimybė skirtingas užduotis vykdyti vienu metu. Mikrovaldikliuose toks paralelinis užduočių vykdymas bandomas imituoti perjunginėjant iš vienos užduoties vykdymo į kitą. Tačiau šiuo atveju šitoks perjunginėjimas mums naudos neatneša, nes bendras paieškos laikas vis tiek bus lygus laikui, kai užduočių perjunginėjimas nėra atliekamas arba net didesnis. Greitaveikos sumažėjimas šiuo atveju įtakojamas tuo, kad užduočių perjungimui taip pat sugaištamas laikas. Antrasis loginių įtaisų pranašumas prieš mikrovaldiklius, tai jų greitis. Loginiai įtaisai lyginant su mikrovaldikliais gali būti taktuojami daug didesniu greičiu.

Algoritmų simuliacijai su programuojamais loginiais įtaisais pasirenkamas Alteros Ciclon II programuojamas loginis įtaisas. Siekiant tiksliau palyginti programuojama logika atliktus bandymus, simuliacijos parametrai parenkami identiškai, kaip ir atliekant bandymus su mikrovaldikliu. Veikimo greitaveika skaičiuojama reikalingais pagrindinio procesoriaus taktavimo taktais. Šitoks matavimo tipas pasirinktas, nes sistemos taktavimo periodas gali būti parenkamas individualiai. Pagrindinis tikslas išsiaiškinti ar programuojamas loginis įtaisas yra pranašesnis už prieš tai aptartą. Realizuotas LFSR metodu pagrįstas paieškos algoritmas, bandymo metu parinktas sekos ilgis 2000 bitų, taip pat fragmento ilgis 80 bitų. Bandymo rezultatai pateikti (4.9) paveiksle.



4.9 pav. FPGA ir mikrovaldiklio realizacijų palyginimas

Kaip galima pastebėti iš (4.9) paveikslo, naudojant mikrovaldiklį LFSR, paremtas algoritmas veikia daug prasčiau nei naudojant FPGA. Naudojant FPGA programuojamąjį įrenginį, atliktų operacijų skaičius nesiekė net 3000, kai tuo tarpu tą pačią užduotį atlikti su mikrovaldikliu reikalinga daugiau nei du milijonai. Deja, dėl ganėtinai sudėtingo koreliacijos ir Euklido algoritmo realizavimo, pasitelkus programuojamą logiką, tikslių ir patikimų duomenų apie Euklido ir koreliacijos algoritmų veikimo greitaveiką gauti nepavyko.

IŠVADOS

Darbe buvo modeliuoti absoliutinės padėties nustatymo, panaudojant pseudo atsitiktinę seką ant keitiklio liniuotės, algoritmai, kurių veikimas pagrįstas koreliacijos, Euklido atstumo skaičiavimu ir atvirkštinio registro principu. Nustatyta, kad,

1. Pagal teisingos padėties nustatymo kriterijų:
 - a. Pavienėms klaidoms atspariausi buvo koreliacijos ir Euklido atstumo algoritmai.
 - b. Optinio jutiklio ir liniuotės žingsnių nesutapimui atspariausias buvo Euklido algoritmas, naudojant vienodus sekos parametrus teisingai aptiktos pozicijos tikimybė iki 30% didesnė.
 - c. Jutiklio ir liniuotės padėties pozicijos nesutapimui labiausiai atsparus buvo Euklido algoritmas. Naudojant vienodus sekos parametrus Euklido atstumu paremto algoritmo teisingai aptiktos pozicijos tikimybė iki 50% didesnė lyginant su koreliacijos metodu paremto algoritmo.
2. Pagal algoritmo realizavimui reikiamų įvykdyti mikrovaldiklio komandų skaičių, mažiausiai imlus buvo koreliacijos algoritmas (naudojant STM bibliotekas). Reikia paminėti, kad LSFR algoritmą realizuoti lygiagrečios duomenų magistralės valdikliu iš principo įmanoma, bet taktinio signalo impulsų skaičiaus prasme, labai neoptimalu. Tuo tarpu srautiniam nuoseklių duomenų apdorojimui skirta programuojama logika architektūrine prasme yra labai gerai tinkama LSFR algoritmo realizavimui.

Buvo pasiūlytas modifikuotas LFSR paremtas algoritmas, kuris atlikus pradinę kalibraciją eliminuoja optinio jutiklio ir liniuotės žingsnių nesutapimo įtaką. Pasiūlytasis metodas pasižymi tuo, kad tikrinant kiekvieno algoritme LFSR generatorių grąžinamą poziciją yra atsižvelgiama į prieš tai eilėje esančio LFSR generatoriaus sugeneruotą reikšmę. Pagal šią reikšmę galima aptikti sekos lūžį dėl žingsnio neatitikimo ir ja dirbtinai pakoreguoti.

LITERATŪROS ŠALTINIAI

1. Encoder Primer [žiūrėta 2014-10-15] prieiga per internetą http://irtfweb.ifa.hawaii.edu/~tcs3/tcs3/0306_conceptual_design/Docs/05_Encoders/encoder_primer.pdf
2. Positional Sensors [žiūrėta 2014-10-20] Prieiga per internetą http://www.electronicstutorials.ws/io/io_2.html
3. Mechanism od absolute rotary encoder [žiūrėta 2014-10-21] Prieiga per internetą http://support.automationdirect.com/docs/absolute_encoders.pdf
4. A. Argeseanu, D. Popovici, O. Cornea ir I. Torac „A new algorithm and device for absolute linear encoders dedicated to long distance applications“ 2008
5. Petriu, E.M. „Absolute position measurement using pseudo-random binary encoding“ 1998
6. Panda, A.K., Rajput, P. ; Shukla, B. „FPGA Implementation of 8, 16 and 32 Bit LFSR with Maximum Length Feedback Polynomial Using VHDL“ 2012
7. Ajane, A.; Furth, P.M. ; Johnson, E.E. ; Subramanyam, R.L. „Comparison of binary and LFSR counters and efficient LFSR decoding“ 2011
8. Goran S. Miljković, Dragan B. Denić, Miroљjub T. Pešić, Miodrag Z. Arsić „Improved pseudorandom absolute position encoder with reliable code reading method“ 2013
9. Dragan B. Denic Goran S. Miljkovic „Code reading synchronization method for pseudorandom position encoders“ 2007
- 10.E. M Petriu „Scanning method for absolute pseudorandom position encoders“ 1988
- 11.Dragan Denić, Goran Miljković, Jelena Lukić, Miodrag Arsić „Pseudorandom position encoder with improved zero position adjustment“ 2012
- 12.Goran Miljković, Dragan Denić, Jelena Lukić, Milan Dinčić „Pseudorandom Position Encoder With Direct Zero Position Adjustment“ 2012
- 13.Denic, D.; Stojkovic, I. „Pseudorandom/natural code converter with parallel feedback logic configuration“ 2010
- 14.Denić, D. Randelovic, I. ; Miljkovic, G. „Recent Trends of Linear and Angular Pseudorandom Encoder Development“ 2006
- 15.Denic, D.; Randelovic, I. „New Type of Position Encoder with Possibility of Direct Zero Position Adjustment“ 2005
- 16.Dragan B. Denić, Ivana S. Randelović, Jelena R. Đorđević, Goran S. Miljković „New Type of Linear and Angular Displacement Transducer Based on Pseudorandom Encoding“ 2005

17.Ajane, A.; Furth, P.M. ; Johnson, E.E. ; Subramanyam, R.L. ; „Comparison of binary and LFSR counters and efficient LFSR decoding algorithm“ 2011