

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

ALGIRDAS KUBILIUS

AUTOMOBILIO SAUGAUS ATSTUMO
PALAIKYMO KOMPIUTERINĖ SISTEMA

Magistro darbas

Vadovas
prof. dr. E. Kazanavičius

KAUNAS, 2014

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

ALGIRDAS KUBILIUS

AUTOMOBILIO SAUGAUS ATSTUMO
PALAIKYMO KOMPIUTERINĖ SISTEMA

Magistro darbas

Vadovas
prof. dr. E. Kazanavičius

Vadovo parašas, data

Recenzentas
doc. A. Liutkevičius

Recenzento parašas, data

Studentas
A. Kubilius

Studento parašas, data

KAUNAS, 2014

TURINYS

1. ĮVADAS.....	5
1.1. Problematika ir aktualumas	5
1.2. Magistrinio darbo tikslas ir uždaviniai	5
1.3. Magistrinio darbo struktūra	6
2. SAUGAUS ATSTUMO PALAIKYMO SISTEMŲ TECHNOLOGIJŲ ANALIZĖ.....	7
2.1. Egzistuojančių atstumo palaikymo sistemų analizė	7
2.2. Atstumo nustatymo sistemų technologijų analizė	10
2.3. Automobilio stabdymo sistemų analizė	10
2.4. Analogiškų sistemų palyginimas.....	12
2.5. Komunikacinių protokolų analizė	13
2.6. Atstumo nustatymo ultragarsu metodų apžvalga	14
2.7. Atstumo įvertinimo principai	14
2.8. Ultragarso bangos sklidimo trukmės nustatymo metodų analizė.....	15
2.8.1. Koreliacinis metodas	15
2.8.2. Maksimumo paieškos metodas.....	16
2.9. Atstumo paskaičiavimas pagal koreliacinės funkcijos rezultata.....	17
2.10. Analizės išvados	17
3. ATSTUMO MATAVIMO KOMPIUTERINĖS SISTEMOS MODELIS.....	19
3.1. Atstumo matavimo sistema	19
3.2. Koreliacijos metodo pritaikymas atstumo paskaičiavimui.....	20
3.3. Matematinis modelis MATLAB aplinkoje	21
3.4. Matematinio modelio patikrinimas	22
3.5. Atstumo nustatymo matavimų fazės algoritmas	23
3.6. Koreliacijos skaičiavimo algoritmas	25
3.7. Koreliacinės sumos kaupimo algoritmas.....	27
3.8. Koreliacijos maksimalios reikšmės fiksavimo algoritmas	28
3.9. Matematinio modelio sudarymo išvados.....	28
4. PRAKTINĖ MATEMATINIO MODELIO REALIZACIJA	29
4.1. Sistemos koncepcijos sudarymas	29
4.2. Ultragarso signalo registravimo techninė įranga.....	30
4.2.1. Matavimų posistemė ir jos veikimas	30
4.3. Eksperimentinio tyrimo metodika	31
4.3.1. Eksperimento priemonės	31
4.3.2. Eksperimento vykdymo schema.....	32
4.3.3. Spartos matavimo metodika	32
4.4. Atstumo matavimo spartos įvertinimas ir palyginimas.....	34
4.5. Atstumo skaičiavimo tikslumo įvertinimas.....	42
4.6. Tyrimo išvados	44
5. IŠVADOS	44
LITERATŪRA.....	45
PRIEDAI.....	48

Paveikslų sąrašas

1 pav. Autonominis Toyota Prius automobilis [4].....	7
2 pav. „City Safety“ sistemos radarų montavimo vieta [6].....	8
3 pav. Atstumo matavimas naudojant atskirų siųstuvo ir imtuvo jutiklių sistemą	15
4 pav. Atstumo matavimas naudojant siųstuvą ir imtuvą suintegruotus į bendrą jutiklį	15
5 pav. Priešingų fazių signalai	16
6 pav. Maksimumo paieškos rezultato iliustracija	16
7 pav. Atstumo matavimo sistemos struktūra	19
8 pav. Registruojamo signalo $y(t)$ pavyzdys	20
9 pav. Matavimai esant 1 m atstumui	22
10 pav. Matavimai esant 1.1 m atstumui	22
11 pav. Ultragarso signalo registravimo modulio veikimo algoritmas	24
12 pav. Pagrindinis koreliacijos algoritmas	26
13 pav. Koreliacinės sumos kaupimo algoritmas.....	27
14 pav. Koreliacijos maksimalios reikšmės fiksavimo algoritmas	28
15 pav. Siūlomos sistemos struktūrinė schema.....	29
16 pav. Ultragarso signalą registruojantis sistemos procesorius.....	30
17 pav. Eksperimento vykdymo schema.....	32
18 pav. Algoritmo vykdymo trukmės matavimo schema	33
19 pav. DSP procesoriuje atliekami skaičiavimai.....	34
20 pav. 220 ilgio sekų koreliavimo palyginimas kai koreliuojama MATLAB ir dsPIC33F platformoje	36
21 pav. Koreliacijos skaičiavimai su neetaloninio signalo poslinkiu	37
22 pav. Fiksuoto ir slankaus taško koreliacijų rezultatų palyginimas.....	38
23 pav. Fiksuoto ir slankaus taško koreliacijų rezultatų palyginimas su poslinkiu	39
24 pav. Slankaus taško koreliacijos pagrindu paskaičiuoti atstumai 1 cm tikslumu	40
25 pav. Fiksuoto taško koreliacijos pagrindu paskaičiuoti atstumai 1 cm tikslumu.....	40
26 pav. Atstumo matavimo vykdymo trukmės slankaus taško koreliacijos atveju	41
27 pav. Atstumo matavimo vykdymo trukmės fiksuoto taško koreliacijos atveju	41
28 pav. Santykinės paklaidos tarp atstumų, paskaičiuotų su slankaus ir fiksuoto taško koreliacijomis	42
29 pav. Ultragarsinės atstumo matavimo sistemos principinė schema	43
30 pav. Ultragarsinė atstumo matavimo sistema	43

Lentelių sąrašas

1 lentelė. Kriterijų įvertinimai.....	12
2 lentelė. Panašių į kuriamą sistemų palyginimas ir įvertinimas	12
3 lentelė. Automobilių pagalbinių sistemų palyginimas ir įvertinimas	13
4 lentelė. Atstumo matavimo sistemos matematinis modelis	21
5 lentelė. Darbe naudojamos DSP platformos parametrai	31
6 lentelė. Skirtingos standartinės koreliacijos realizacijos.....	34
7 lentelė. Vykdymo trukmių palyginimo lentelė	42

1. Įvadas

1.1. Problematika ir aktualumas

Automobilis šiuo metu yra viena iš pagrindinių transporto priemonių šiuolaikiniam žmogui. Kiekvienais metais jų pagaminama vis daugiau, nenuostabu, kad ir nuperkama daugiau. Remiantis Production Statistics [1] duomenimis, nuo 1999 iki 2011 metų pagamintų automobilių kiekis išaugo nuo 40 mln. iki 59 mln., o tai reiškia beveik apie pusantro karto padidėjimą. Taigi, neišvengiamai didėja ir eismo įvykių skaičius.

Eismo įvykių padariniai gali būti nuo menkų įbrėžimų stovėjimo aikštelėje, apie kuriuos net nepranešama draudimo įmonėms ir policijai, iki eismo įvykių, kurių metu yra mirtinai sužalojami žmonės ir patiriami dideli materialiniai nuostoliai. Kartais ir smulkus eismo įvykis gali sukelti labai daug nuostolių ir blogų padarinių. Kaip pavyzdį galima paminėti Vokietijos greitkelį, dar vadinamą autobanu. Ten menkas eismo įvykis gali sukelti susidūrimą su keliomis dešimtimis automobilių, ko pasekoje išauga ne tik tikimybė būti mirtinai sužalotam, bet ir materialiniai nuostoliai dėl sudaužytų automobilių.

Remiantis K. Rumar veikalu „The Role of Perceptual and Cognitive Filters in Observed Behavior," *Human Behavior in Traffic Safety*“, galima teigti, jog pagrindinė eismo įvykių priežastis yra žmogiškasis faktorius. Jis sudarė diagramą, remdamasis Jungtinės Karalystės ir Jungtinių Amerikos Valstijų eismo įvykių priežasčių duomenimis [2]. Iš jos galima spręsti, jog vien tik žmogiškasis faktorius nulemia beveik du trečdalius eismo įvykių. Pridėjus kelio bei automobilio būklės įtaką, eismo įvykių, dėl kurių (iš dalies) kaltas yra vairuotojas, skaičius siekia net apie 90 procentų.

Dėl šios priežasties kuriama įvairių mechaninių, elektroninių ir mechatroninių sistemų, kurios padėtų lengviau pastebėti pėsčiuosius bei važiuojančius kitus eismo dalyvius, greičiau ir saugiau reaguoti į pavojus ir kitaip sumažinti žmogaus „indėlių“ prisidedant prie skaudžių nelaimių.

Darbo aktualumą nusako Europos Sąjungos direktyva 2010/40/ES, kurioje pabrėžiama, kad didėjant kelių transporto srautų apimčiai, kuri susijusi su Europos ekonomikos augimu ir didėjant piliečių judumui negalima apsiriboti vien tradicinėmis priemonėmis. Svarbus vaidmuo tenka inovacijoms. Direktyvoje teigiama:

„Intelektinės transporto sistemos (ITS) yra pažangios prietaikos, kuriomis, nors jos neturi paties intelekto, siekiama tiekti inovacines paslaugas, susijusias su įvairiomis transporto rūšimis ir eismo valdymu ir sudaryti sąlygas įvairiems naudotojams gauti geresnę informaciją ir saugiau, geriau koordinuoti bei „išmaniau“ naudotis transporto tinklais“

Atliekamas darbas priklauso vienai iš ITS prioritetinių sričių, apibrėžtų direktyvoje, t. y. „ITS kelių saugos ir saugumo prietaikos“.

1.2. Magistrinio darbo tikslas ir uždaviniai

Pagrindinis magistrinio darbo tikslas – sukurti automobilio saugaus atstumo palaikymo kompiuterinę sistemą, apimančią atstumą skaičiuojantį algoritmą ir jam realizuoti reikalingą techninės įrangos struktūrą, jos praktinę realizaciją, kuri stebi automobilio atstumą iki priešais esančių objektų (automobilių).

Darbo tikslui įgyvendinti keliami sekantys uždaviniai:

1. Atlikti egzistuojančių sistemų technologijų analizę ir jose naudojamų metodų ir būdų atstumo nustatymui iki kliūties.

2. Atlikti komunikacinių protokolų analizę. Analizuojami komunikaciniai protokoliai naudojami automobilių pramonės gamintojų.
3. Sudaryti atstumo nustatymo algoritmą.
4. Sudaryti atstumo matavimo matematinį modelį.
5. Atlikti sistemos praktinę realizaciją. Atlikti sumodeliuotos sistemos realizaciją. Sudaryti algoritmo realizaciją konkrečia programavimo programa C/C++ ir įrašyti į sistemos DSP valdiklį.
6. Atlikti sistemos matematinio modelio praktinį įdiegimą ir jo eksperimentinį tyrimą.

1.3. Magistrinio darbo struktūra

Pirmoje šio darbo dalyje suformuluota atliekamo darbo problematika ir aktualumas bei sprendžiami uždaviniai.

Antroje dalyje atlikta egzistuojančių atstumo nustatymo sistemų technologijų analizė atsižvelgiant į tai, kaip sistemose panaudojama ultragarso technologija. Analizuojami komunikaciniai protokoliai, kurie leistų kuriamai sistemai lengvai integruotis į automobilyje jau esančią kompiuterinę sistemą.

Trečioje dalyje pateiktas teorinis-matematinis sistemos veikimo modelis. Pateiktas atstumo matavimo algoritmas ir matematinio modelio realizacija su patikrinimu MATLAB aplinkoje.

Ketvirtoje darbo dalyje aprašoma pasirinkta DSP architektūra. Pateiktos eksperimentų atlikimo metodikos ir eksperimentinių tyrimų rezultatai.

Prieduose pateikta matematinio modelio programa MATLAB aplinkoje ir DSP procesoriaus programinis kodas.

2. Saugaus atstumo palaikymo sistemų technologijų analizė

Šios darbo dalies tikslas yra ištirti, kokios šiuo metu kuriamos ir jau sukurtos automobilių saugaus atstumo palaikymo, automobilio aplinkos stebėjimo ar panašios sistemos, kurios iš dalies, tam tikromis funkcijomis, yra panašios į kuriamą sistemą. Vienas iš svarbiausių keliamų uždavinių yra atstumo iki priekyje esančio objekto apskaičiavimas, todėl toliau bus analizuojamos sistemos, kurių pagrindinis arba vienas iš pagrindinių tikslų yra atstumo skaičiavimas. Jeigu sistema sudaryta ir iš kitų sudedamųjų dalių – modulių, bus nagrinėjama, kaip šie moduliai yra integruojami į visumą ir kaip panaudojami skaičiavimų duomenys siekiant bendro rezultato.

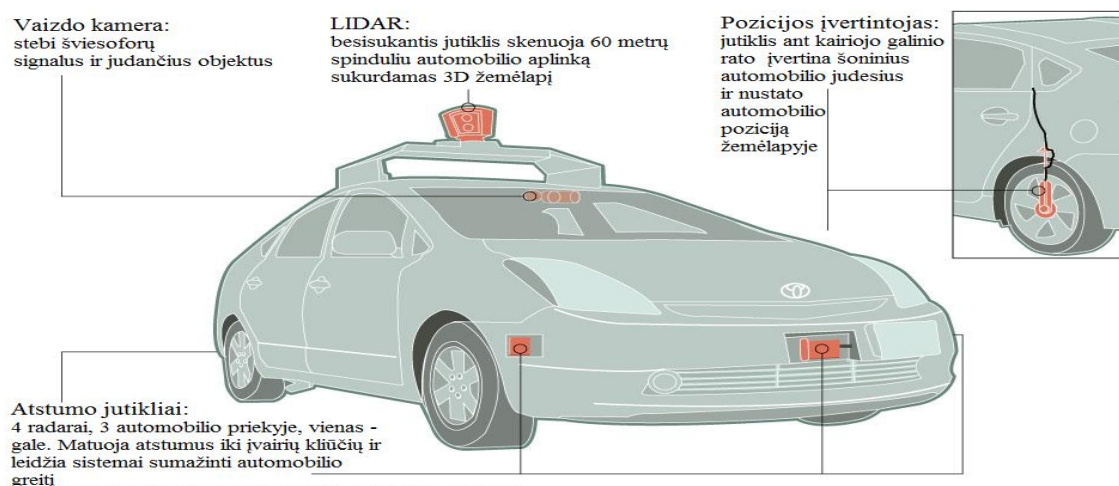
Tam, kad patikrinti, ar jau esančios sistemos ir sprendimai tinkamai atlieka iškeltus uždavinius, išskiriami svarbiausi kriterijai, į kuriuos derėtų atsižvelgti:

- kuo didesnis atstumas, kuriuo aptinkamas priekyje esantis automobilis ar panašus objektas – mažiausias turėtų būti bent 1 metras;
- kuo didesnis automobilio judėjimo greitis, kuriam esant, sistema aktyvi. Stovėjimo aikštelėse greitis būna apie 20 km/h, mieste maksimalus leistinas greitis yra 50 km/h, todėl mažiausias greitis sistemai turėtų būti bent 20 km/h;
- kuo mažesnė aplinkos sąlygų įtaka sistemos patikimumui. Sistema turėtų gerai veikti ir šviesiu, ir tamsiu paros metu, taip pat lyjant, sningant ir esant giedram orui;
- ar yra galimybė įsigyti atskirai. Vartotojas turi turėti galimybę įsigyti tokią sistemą atskirai ir įsidiegti į savo automobilį;
- kuo mažesnė kaina, neviršijant 1000 litų.

2.1. Egzistuojančių atstumo palaikymo sistemų analizė

Šiame skyrelyje analizuojami žinomiausių automobilių kompanijų sprendimai.

Google sprendimas. Saugaus eismo plėtojimo ir eismo įvykių mažinimo pavyzdžiu gali būti Google kuriama sistema. 2010 metais „The New York Times“ pasirodė straipsnis apie tai, kad Google smarkiai patobulino Toyota Prius automobilį [3] (1 pav.).



1 pav. Autonominis Toyota Prius automobilis [4]

Pirmame paveiksle matyti, jog automobilyje įrengta įvairių jutiklių. Vienas iš svarbiausių yra lazeris (LIDAR), pritvirtintas ant automobilio stogo. Jis sukuria 3D atvaizdą aplinkos apie 60 metrų spinduliu nuo lazerio aplink visą automobilį. Lazeris palygina sudarytą atvaizdą su prieš tai sudarytu

atreipdamas dėmesį į judančius objektus (žr. 1 pav. atstumo jutikliai) . Taip pat įvertinami papildomi duomenys gaunami iš automobilio ratų daviklių. Lazerio ir ratų daviklių derinimas leidžia automobiliui judėti nustatytu maršrutu kelių centimetrų tikslumu, ko nebūtų galima pasiekti naudojant vien tradicines GPS sistemas. Google atstovai teigia, kad tokia autonominių automobilių technologija gali neatsipirkti dar daugelį metų. Dėl to tokie tyrimai, bandymai yra gana rizikingi.

Volvo sprendimas. 2007 metais pasirodė straipsnis [5] apie tai, kad Volvo sukūrė sistemą, pavadinimu „City safety“ („Miesto saugumas“), kurios veikimo principas: jei vairuotojas ruošiasi susidurti su automobiliu, esančiu priešais, ir nereaguoja, automobilis sustabdomas.

Jei priešais esantis automobilis staiga stabdo, „City safety“ paruošia stabdžių sistemą efektyvesniam stabdymui taip, kad nors ir silpnai spustelėjus pedalą, stabdoma bus smarkiai, o jei susidūrimas neišvengiamas, automobilis sustabdomas hidraulinės sistemos pagalba. Ši sistema veikia vienodai gerai nepriklausomai nuo paros meto.

„City safety“ sistema yra aktyvi, kai automobilio greitis yra iki 30 km/h. Jeigu greitis yra mažesnis nei 15 km/h, sistema padeda visiškai išvengti susidūrimo.

Sistemos „akis“ yra optinis radaras, įmontuotas prie priekinio stiklo, galinio vaizdo veidrodėlio aukštyje (2 pav.).



2 pav. „City Safety“ sistemos radarų montavimo vieta [6]

Jis gali aptikti automobilius, esančius 6-8 metrų atstumu [7], kurie arba stovi vietoje, arba važiuoja ta pačia kryptimi. Sistema 50 kartų per sekundę apskaičiuoja stabdymo greitį ir jėgą. Jeigu apskaičiuota jėga viršija nustatytą ribą be vairuotojo įsikišimo, nusprendžiama, kad susidūrimas neišvengiamas, ir tokiu atveju aktyvuojami stabdžiai.

Vienas iš didžiausių šios sistemos trūkumų, atsižvelgiant į kriterijus, yra neatlaidumas oro sąlygoms. Technologinis paketas, kurį sudaro ir susidūrimo išspėjimo, ir išlikimo kelio juostose bei pėsčiųjų aptikimo sistemos, kainuoja 2100 JAV dolerių [8] (apie 5400 Lt).

BMW sprendimas. 2010 metais BMW įdiegė sistemą „Connected Drive“, išpėjančią apie susidūrimą ir automatiškai stabdančią [9]. Jeigu priešais važiuojantis automobilis staiga stabdo, pirmiausia vairuotojui rodomas optinis perspėjimas. Jei vairuotojas nereaguoja, aktyvuojamas garsinis signalas, turintis patraukti dėmesį. Jei ir tada nereaguojama, automobilis pradeda pats stabdyti.

Patobulinta šios technologijos versija „Connected Drive Connect“ įdiegta į 5 serijos BMW automobilį [10]. Sistema naudoja radarą, kameras, lazerinius skaitytuvus ir ultragarsinius atstumo jutiklius, kad matytų aplinką. Automobilis gali vairuoti pats, turėdamas tik informaciją apie kelią, jo juostas, kuri įrašyta į kompiuterį centimetro tikslumu. Automobilis netgi sugeba apvažiuoti kitą, lėtesnį automobilį, važiuojantį toje pačioje juostoje, ir sugrįžti į buvusią juostą. Sistemų kūrėjai išpėja, kad ji nėra skirta visiškai pakeisti vairuotoją, o tik padėti suvaldyti automobilį kritinėse situacijose.

Matyti, kad pagalbinio vairavimo sistemos kuriamos gana ilgai, tačiau visuomenei kol kas nėra pateikiamos. Nors apie kainą neskelbiama, iš to, kad sistemos pagalba atliekami sudėtingi skaičiavimai, o ir patys gamintojai dar bent 10 metų nesiruošia pradėti masinės gamybos, galima daryti išvadą, jog įranga yra brangi. Automobilyje diegiami kompiuteriai, atliekantys skaičiavimus,

atima vairavimo komfortą. Išvelgiamas aukštas sistemos patikimumas, nes naudojama daug ir skirtingų jutiklių: jei gamtos stichijos, pavyzdžiui, lietus, blokuotų vaizdinius jutiklius, gerai veiktų ultragarsu parenti, o ilgesnius atstumus, kurie galėtų būti per dideli ultragarsiniam signalui, puikiai ir greitai įveikia lazeriniai.

Ford sprendimas. Ford kompanija sukūrė labai panašią sistemą į Volvo, kuri vadinasi „Ford Active City Stop“ [11], [12]. Sistema paremta LIDAR technologija, kuri naudoja optinį jutiklį, jog aptiktą kliūtį priekyje automobilio ir esant ne didesniai greičiui nei 30 km/h sustabdytų automobilį pažeidus saugų atstumą. Maksimalus atstumas, ties kuriuo sistema galėtų sustabdyti automobilį, yra 7,6 metro, tai maždaug atitinka stabdymo kelių idealiomis sąlygomis esant 30 km/h greičiui, dėl to, sistema yra aktyvi iki 30km/h greičio. Jei vairuotojas kaip nors įsikiša sistemai suveikus, pavyzdžiui, vairuotojas staiga pasuka vairą, sistemą išsijungia ir leidžia vairuotojui imtis tolesnių veiksmų stengiantis išvengti susidūrimo.

Sistema, pastebėjusi, jog priešais esančios kliūtis greitis staiga sumažėjo, vairuotoją perspėja garsiniu signalu, jei iki susidūrimo lieka labai mažas atstumas, ant priekinio stiklo pradeda mirksėti raudona lemputė, o jei vairuotojas vis vien nesureaguoja į tai, tuomet sistema pasirošia agresyviai stabdymui. Sistema seka, kaip vairuotojas elgiasi, jei jis staiga atleidžia akseleratoriaus pedalą, sistema pradeda stabdyti, jei vairuotojas tuo metu nuspaudžia stabdymo pedalą, sistema stabdo pilnu pajėgumu. Vairuotojas turi galimybę tą sistemą išjungti, bet kiekvieną kartą užvedant automobilį, sistema vėl automatiškai įsijungs.

Sistemos trūkumai: jautrumas oro sąlygoms ir išoriniams veiksniams (purvas, lapai). Šios sistemos įsigijimas yra apribotas, ji negali būti įdiegta į visų tipų automobilius. Tokią sistemą galima pasirinkti kaip priedą renkantis naują automobilį, kaina be PVM – 4355 litai.

Mercedes-Benz sprendimas. Mercedes-Benz sukurta Pre-Safe sistema atlieka vairuotojo įspėjimo ir automatinio stabdymo veiksmus [13]. Pre-Safe yra sujungta su Brake Assist ir Electronic Stability Program (ESP) sistemomis, kurios kartu milisekundžių tarpe aptinka kritiškas situacijas vairavimo metu ir siunčia signalus į valdymo įtaisus. Patobulinta sistema naudoja įvairius jutiklius, kad būtų galima veikti nelaukiant vairuotojo reakcijos. Sistema pritaikyta greitkeliams, kur automobilių greičiai bei atstumai tarp jų didesni.

Jeigu vairuotojas nereaguoja į tris garsinius įspėjimus apie gresiantį susidūrimą, likus apie 1,6 sekundėms iki apskaičiuoto susidūrimo Pre-Safe įjungia stabdžius, naudodama apie 40 % jų jėgos (maždaug 4 m/s^2). Jei vairuotojas sureaguoja, stabdžius gali nuspausti su visa 100 procentų jėga. Priešingu atveju, likus apie 0,6 sekundės iki susidūrimo, stabdžiai įjungiami naudojant visą jų galią, taip žymiai sumažinant susidūrimo smūgį. Be to, priklausomai nuo įdiegtos kitos įrangos, gali būti uždaromi priekiniai langai, įtempiami saugos diržai, pareguliuojamas sėdynės atstumas nuo vairo tam, kad būtų optimalus kampas išsiskleidžiant oro pagalvei.

Ši sistema veikia esant 30-200 km/h važiavimo greičiui, o jeigu greitis mažesnis nei 70 km/h, atskiria artėjimą prie stovinčių automobilių eilės. Naudojami du radarai, siekiantys 30 metrų, ir vienas ilgo nuotolio radaras, siekiantis iki 200 metrų (kai kurių šaltinių teigimu – iki 150 metrų [14]).

Parkavimo sistemos. Parkavimo sistema [15] yra prietaisas, skirtas nustatyti momentiniam atstumui tarp mašinos ir kliūtis. Sistemą sudaro ultragarsiniai jutikliai ir valdymo prietaisas. Jie veikia palyginti nedideliu atstumu, apie vieną metrą, ir yra išdėstyti automobilio šonuose. Tokios sistemos naudoja garsinius signalus, jog įspėtų vairuotoją apie aptiktą artimą kliūtį: kuo arčiau kliūtis, tuo dažniau suveikia garsinis signalas. Remiantis „BMW“ sukurta sistema [16], kai iki kliūtis lieka tik 30 centimetrų atstumas, garsinis signalas būna pastovus. Taipogi „BMW“ siūlo ne tik garsinius signalus įspėjimui, tačiau dar naudoja ir vizualines priemones konsolėje įmontuotuose vaizduokliuose. Aišku, „BMW“ nėra vieninteliai, siūlantys tokią įrangą, ją gali pasiūlyti beveik visi šiuo metu esantys automobilių gamintojai.

Tokios rūšies sistemos, kainuojančios kelis šimtus litų ir prieinamos daugeliui pirkėjų, yra patogios: nereikia šalia vežtis kompiuterių ar kitokios įrangos. Be to, jos veikia nedideliais atstumais, kas yra labai svarbu, stengiantis pastatyti automobilį kaip įmanoma arčiau sienų ar kitų automobilių, ir atstumas skaičiuojamas pakankamai greitai. Deja, nedidelis kliūčių aptikimo atstumas netinkamas važiuojančiam automobiliui, nes net ir esant palyginti nedideliui greičiui (apie 30-40 km/h), ir ypač

esant slidžiai kelio dangai, stengiamasi išlaikyti didesnę nei vienas metras atstumą iki priekyje esančio automobilio.

2.2. Atstumo nustatymo sistemų technologijų analizė

Kuriamos sistemos pagrindinės funkcijos yra atstumo iki kliūties skaičiavimas, ir, esant atitinkamoms sąlygoms, automobilio sustabdymas, dėl to nagrinėta, kokie gali būti tinkami metodai atstumo apskaičiavimui ir stabdymui. Atstumo skaičiavimas paprastai paremtas įvairios prigimties bangų sklidimo ir atsispindėjimu nuo kliūties, arba aidu. Labiausiai paplitę metodai naudoja garso bangas iš ultragarsinių dažnių diapazono, ir elektromagnetines bangas iš radijo, infraraudonosios, matomos bei ultravioletinės šviesos diapazonų. Toliau išanalizuotos pagrindinės atstumo skaičiavimo bei stabdžių technologijos.

Elektromagnetinės bangos. LIDAR (*Light Detection And Ranging*) [17], kariniame kontekste dažniau vadinamos LADAR (*Laser Detection And Ranging*), yra optinės nuotolinio objektų tyrimo technologijos. Paprastai skaičiuojamas atstumas iki objektų, nors gali būtų tiriamos ir kitos charakteristikos: judėjimo greitis, kryptis ar vidinė sandara. Veikimo principas paremtas infraraudonosios, matomos arba ultravioletinės šviesos (elektromagnetinių bangų, kurių sklidimo dažnis pakliūva į šį diapazoną) spinduliavimu ir atsispindėjimu, kai skaičiuojama atspindžio grįžimo trukmė ir žinant šviesos sklidimo greitį apskaičiuojamas atstumas. Toks skaičiavimo būdas netinkamas mažiems atstumams skaičiuoti dėl labai didelio šviesos greičio, ir tuomet naudojami trianguliacijos metodai. Jeigu vis tik nusprendžiama naudoti lazerį nedidelių atstumų skaičiavimams, tam reikia kompiuterių, kurie turi pakankamai resursų atlikti labai daug skaičiavimų per labai trumpą laiko tarpą, mat jei pagrindinio procesoriaus darbo periodas bus ilgesnis negu šviesos atspindžio keliavimo trukmė, ji tiesiog nebus fiksuojama. Būtent dėl to, jog naudojant lazerį atstumo skaičiavimui reikia galingo kompiuterio ar serverio, šio metodo negalime pasirinkti kuriamai sistemai dėl įrangos ir lėšų įrangai įsigyti trūkumo.

Garso bangos. Panašus į anksčiau aprašytą metodą yra ultragarsinis objektų aptikimas, sutrumpintai vadinamas SONAR (*SOund Navigation And Ranging*) [18]. Šis metodas taip pat pasitelkia garso bangų, kurių dažnis svyruoja nuo 20 kHz iki 200 MHz ir daugiau, sklidimą ir atsispindėjimą nuo nutolusių objektų. Paprastai sklidimo terpė būna vanduo, nors taip pat sėkmingai gali būti ir oras, ir kietieji kūnai, kurių vidinė struktūra būna tiriami, ar net žmogus. Kadangi garso bangų sklidimo greitis (ore) žymiai mažesnis nei šviesos, ultragarsinis atstumo skaičiavimas gali būti taikomas mažesniems atstumams. Čia susiduriama su priešinga problema: kai norima apskaičiuoti atstumą tarp dviejų judančių objektų vienas kito atžvilgiu, ir jeigu jie dar daug nutolę vienas nuo kito, atsiranda Doplerio efektas, kuris paaiškina garso bangų sutankėjimą arba išretėjimą (priklausomai nuo to, ar objektai tolsta vienas nuo kito, ar artėja). O tai sukelia garso dažnio pokytį, kartu ir sklidimo greitį ir galiausiai lemia netikslius skaičiavimų rezultatus.

2.3. Automobilio stabdymo sistemų analizė

Analizuojant panašias atstumo palaikymo sistemas analizuotina ir bene svarbiausia automobilio atstumo palaikymo sistemos dalis – stabdžiai. Stabdžių veikimas pagrįstas elektronika arba hidraulika. Kiekvienas iš šių tipų aprašomi atskirai.

Elektroniniai stabdžiai. Manoma, jog tokio tipo stabdžių sistemos greitai laiku perims rinką iš hidraulinių stabdymo sistemų, kadangi remiantis „Siemens“ korporacija, jų kuriama „Electronic Wedge Brake“ (EWB) [19] bus kur kas efektyvesnė, kainuos mažiau, dirbs greičiau ir žmogus galės atlikti stabdymo manevrus kur kas saugiau. Sistema yra gan paprasta, ją sudaro [20] stabdžių diskai, stabdžių kaladėlės, kurias prispaudžia du elektriniai varikliukai prie stabdžių disko, pagrindinis mikrovaldiklis ir atsarginis elementas. Ant kiekvieno stabdžių disko bus pritvirtinti keturi davikliai, kurių dėka bus galima žinoti automobilio greitį ir kitas reikalingas stabdymui charakteristikas. Nuspaudus stabdžių pedalą, pagal tai, kiek stabdžių pedalus bus nuspaustas, tokia jėga stabdžių

diskus suspaus elektriniai varikliai, davikliai užtikrins, jog vairuotojas nenuspaustų per daug stabdžių pedalo nereikiamu metu, tačiau jei tai būtų kritinė situacija, kai vairuotojas stipriai nuspaudžia stabdžių pedalą, tuomet sistema suveiktų pilnai blokuodama ratus, bet su stabdžių antiblokavimo sistema (ABS, *Anti-lock braking system*) būtų stabdoma efektyviau, tai yra ratai neužblokuojami ir stabdomas automobilis didžiausiu galimu efektyvumu. Jei automobilio elektros sistema sugestų ir staiga nebūtų pakankamo maitinimo stabdžiams, tam yra pastatytas atskiras elementas, skirtas tokioms klaidoms panaikinti ir užtikrinti, jog tokia stabdžių sistema veiks ne tik efektyviai, bet ir saugiai. Taipogi tokio tipo stabdymo sistemą turinčiame automobilyje būtų kur kas lengviau įdiegti kuriamą produktą, kadangi užtektų pagrindiniam stabdymo mikrovaldikliui nusiųsti signalą, jog pristabdyti ar stabdyti automobilį.

Hidrauliniai stabdžiai. Hidraulinės stabdymo sistemos [21] veikia tokiu principu: tam tikra jėga yra panaudojama vienoje vietoje ir ta jėga yra perduodama į kitą tašką pasinaudojus uždaroje sistemoje esančiu nesuspaudžiamu skysčiu, tam dažniausiai yra naudojamas hidrolis. Daugelyje sistemų, jėga yra padidinama, jog išgauti didesnę stabdymo galią. Tarkime, jog turime du cilindrus, pripildytus tepalo ir tarpusavyje sujungtus vamzdeliu. Vieno cilindro diametras yra apie 5 centimetrus, kito, apie 15 centimetrų. To pasekoje, paviršiaus plotas antrajame cilindre yra apie 9 kartus didesnis ir pirmajame cilindre panaudota jėga antrajame pasirodys 9 kartus didesnė. Taigi hidraulinę stabdymo sistemą sudaro stabdymo pedalas, stūmoklis, pagrindinis agregatas, kuriame yra stūmokliai, kurių dėka yra spaudžiamas skystis, stabdžių diskai, stabdžių kaladėlės. Nuspaudus stabdžių pedalą, stūmokliai spaudžia skystį, kuris padidinta jėga per kitą stūmoklį ir yra perduodamas žarnelėmis iki stabdžių kaladėlių ir jos yra prispaudžiamos prie stabdžių disko. Pritaikant tokio tipo stabdymo sistemas kuriamam produktui yra sudėtingiau, nei su elektroniniais stabdžiais, kadangi čia reikia variklio, kuriam siųstų signalą mikrovaldiklis su tam tikra jėga stabdymui ir būtų nuspaudžiamas stūmoklis. Pirmą tokia sistema buvo sukurta 1918 m. ir nuo to laiko nelabai kuo pakito, principas išliko tas pats. Remiantis „Siemens“, hidraulinių sistemų eksploatacijos, kaip ir pačios sistemos, kaina yra kur kas didesnė negu elektrinių stabdžių. Hidraulinė sistema nėra tokia efektyvi kaip elektroninė.

2.4. Analogiškų sistemų palyginimas

Pagal pasirinktus kriterijus buvo nusistatyti įvertinimai, kurių pagalba kiekviena analogiška ar panaši sistema buvo įvertinta, tam, jog pažiūrėti, ar tikrai reikia apsiimti kurti panašią sistemą. 1 lentelėje pateikti kriterijų įvertinimai su reikšmėmis, ties kuriomis sistemai suteikiamas atitinkamas įvertinimas atitinkamam kriterijui.

1 lentelė. Kriterijų įvertinimai

Kriterijus	Reikšmė	Įvertinimas
Atstumas iki priekyje esančios kliūtis	< 1 m	Labai mažas (1)
	1-3 m	Mažas (2)
	3-6 m	Vidutinis (3)
	6-9 m	Didelis (4)
	> 10 m	Labai didelis (5)
Važiavimo greitis, kai sistema aktyvi	< 10 km/h	Labai mažas (1)
	10-20 km/h	Mažas (2)
	20-30 km/h	Vidutinis (3)
	30-40 km/h	Didelis (4)
	> 40 km/h	Labai didelis (5)
Aplinkos sąlygų įtaka sistemos veikimui	Neveikia	Labai didelė (1)
	Beveik neveikia	Didelė (2)
	Dalinai veikia	Vidutinė (3)
	Veikia su nežymiomis paklaidomis	Maža (4)
	Puikiai veikia	Labai maža (5)
Ar galima įsigyti atskirai	Negalima įsigyti	Negalima (0)
	Galima įsigyti	Galima (2)
Kaina	> 2400 Lt	Labai didelė (1)
	1200-2400 Lt	Didelė (2)
	700-1200 Lt	Vidutinė (3)
	400-700 Lt	Maža (4)
	< 300 Lt	Labai maža (5)

Toliau yra pateikiamos dvi lentelės, 2-oje bus įvertintos sistemos, kurios yra panašios į norimą kurti sistemą, o 3-ioje – analogiškų sistemų įvertinimas ir palyginimas.

2 lentelė. Panašių į kuriamą sistemų palyginimas ir įvertinimas

Kriterijus	Google sprendimas	Parkavimo sistemos
Atstumas iki priekyje esančio automobilio	Labai didelis (>10 m) (5)	~1 m (1)
Važiavimo greitis, kai sistema aktyvi	Labai didelis (>40 km/h) (5)	< 10 km/h (1)
Aplinkos sąlygų įtaka	Labai maža (5)	Puikiai veikia (5)
Ar galima įsigyti atskirai	Negalima (0)	Galima (2)
Kaina	Labai didelė (> 3000 Lt) (1)	Keli šimtai litų (4)
Bendras įvertinimas	21	18

3 lentelė. Automobilių pagalbinių sistemų palyginimas ir įvertinimas

Kriterijus	Volvo sprendimas	Ford sprendimas	Mercedes-Benz sprendimas
Atstumas iki priekyje esančio automobilio	~7 m (4)	~7,6 m (4)	<200m (5)
Važiavimo greitis, kai sistema aktyvi	30 km/h (4)	30 km/h (4)	30-200 km/h (4)
Aplinkos sąlygų įtaka	Maža (4)	Maža (4)	Maža (4)
Ar galima įsigyti atskirai	Negalima (0)	Negalima (0)	Negalima (0)
Kaina	Labai didelė (~5400 Lt) (1)	Labai didelė (~5300 Lt) (1)	Apie 54000 Lt (visa saugumo įranga) (1)
Bendras įvertinimas	18	17-18	18-19

Galiausiai pagrindinį dėmesį reikia atkreipti į analogiškas sistemas, kadangi kuriama sistema turės konkuruoti su būtent tokio pobūdžio sistemomis.

2.5. Komunikacinių protokolų analizė

Kuriant sistemą laikoma, jog automobilis į kurį bus diegiama sistema turi įdiegtą variklio valdymo įrenginį (angl. *ECU – Engine Control Unit*). Anot [22] šaltinio, ECU yra elektroninis valdymo įrenginys (angl. *ECU – Electronic Control Unit*), kuris valdo vidaus degimo variklio mechaninius įrenginius, užtikrinančius optimalų variklio veikimą. Tai tėra tik pradinis ECU apibrėžimas susijęs su šių įrenginių įsigalėjimu automobiliuose. Jų pradžia galima laikyti 1980 metus, kai buvo kuriami hibridiniai skaitmeniniai-analoginiai ECU.

Intensyviai vystantis elektroninei technikai pradėti kurti šiuolaikiniai modernūs ECU. Jie nuo savo pirmtakų skiriasi tuo, kad naudoja mikroprocesorius, kurie apdoroja analoginius signalus gaunamus iš automobilio variklio jutiklių realiu laiku. Šiuolaikinių ECU valdymas neapsiriboja vien tik variklio valdymu, tačiau apima ir kitas automobilyje išvystytas elektrines posistemas tokias kaip kruizo valdymas, pavarų dėžės valdymas, stabdžių sistemos slydimui sumažinti valdymas ir pan. Šios ir panašios posistemės apjungiamos tarpusavyje ir susiejamos su vienu pagrindiniu ECU. Štai čia ir išauga komunikacinių protokolų poreikis, kad tarpusavyje susietos posistemės galėtų komunikuoti tarpusavyje.

Kuriamos sistemos integracija į automobilį pareikalauja, kad sistema galėtų komunikuoti su automobilio elektrine stabdžių valdymo posisteme ir su kitomis posistemėmis, nes reikalinga išgauti duomenis apie automobilio judėjimo parametrus. Taigi, dabartiniu metu egzistuojantys komunikacinių protokolų standartai yra šie: CAN [23], LIN [24], ISO-9141 [25], J1939 [26], KWP2000 [27] ir kiti.

CAN protokolas. Transporto priemonės protokolo standartas CAN [23] (angl. *Controller Area Network*) leidžia elektriniams įrenginiams, turintiems mikrovaldiklius, komunikuoti tarpusavyje be pagrindinio kompiuterio. CAN yra „žinutėmis“ paremtas protokolas, sukurtas ir pritaikytas būtent automobiliams. Šiuolaikinis automobilis gali turėti apie 70 ECU įvairioms posistemėms. Baziniu procesoriumi išlieka tas pats variklio valdymo įrenginys (Engine CU). Kiti naudojami pagalbinėms sistemoms (pavarų dėžėms, oro pagalvėms, ABS, kruizui, vairo stiprintuvui, garso sistemai, langams, durelėms, veidrodėliams pareguliuoti, baterijoms ir jų krovimo sistemoms skirtoms hibridiniams ar elektriniams automobiliams ir pan.) valdyti. Taigi, žiūrint iš technologinės pusės CAN yra daugelio

valdančių įrenginių (angl. *multi-master*) nuoseklaus transliavimo magistralės standartas apjungiantis elektroninius valdymo įrenginius (ECU).

LIN protokolas. Nuoseklaus tinklo protokolo standartas LIN [24] naudojamas komunikacijai tarp komponentų automobilyje. Šio protokolo sukūrimą paskatino pigaus ir nuoseklaus tinklo poreikis, kai technologijų naudojimas automobilyje taip pat išaugo, o CAN protokolą buvo pernelyg sunku pritaikyti kiekvienam komponentui esančiam automobilyje. Šio transliuojančio protokolo suteikiama tinklo topologija leidžia, kad tinkle būtų vienas valdantysis (angl. *master*) komponentas ir iki 16 jam pavaldžių (angl. *slave*) komponentų.

Protokolo taikymų segmentai: automobilio stogas, vairas, sėdimos vietos automobilyje, variklis (jutikliai ir varikliai), klimato kontrolė, durų valdymas.

Protokolo teigiamos savybės yra šios: paprastas naudoti, rinkoje plačiai parduodami komponentai su šio protokolo palaikymu, pigesnis nei CAN ar kitos komunikavimo magistralės, su šiuo protokolu automobilio elektroninės sistemos yra patikimesnės, nereikalingas licencijos mokestis. LIN gera alternatyva, kur reikalinga maža sistemos kaina duomenų perdavimo spartos sąskaita.

Kiti protokolai, jų standartai. Standartas ISO-9141 [25] specifikuoja reikalavimus informacijos mainams tarp ECU ir tinkamo diagnostinio matavimo įrenginio, kad palengvintų automobilio ECU apžiūrą, matavimus, sistemos diagnozę ir sistemos parametrų reguliavimą.

Protokolas SAE (angl. *Society of Automotive Engineers*) J1939 [26] yra atviras standartas komunikaciniams tinklams komercinėse transporto priemonėse. Šis standartas yra paremtas CAN technologijos pagrindu.

Komunikacinis protokolas KWP2000 [27] naudojamas diagnostinėms sistemoms (OBD – angl. *On-board diagnostics*). Protokolas standartizuotas kaip ISO 14230 (angl. *International Standard Organization*) standartas. Šio protokolo fizinis lygmuo OSI modelyje yra identiškas ISO 9141 standartui. Taip pat suderinamas ir su CAN (ISO 11898).

Yra ir kitų protokolų taikomų transporto priemonėse, tačiau jie labiau orientuoti į tam tikras transporto priemonių kategorijas, pavyzdžiui, krovininiai automobiliai, lengvieji ir kt. Pavyzdžiui, ISO 9141 taikomas diagnostiniams įrenginiams.

2.6. Atstumo nustatymo ultragarsu metodų apžvalga

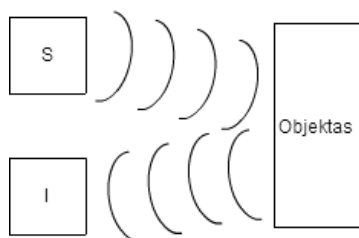
Atstumo matavimas ultragarsu plačiai taikomas pramonės automatikoje, robotikoje, automobiliuose (parktronikai). Pramonės automatikos pavyzdžiais gali būti skysčio lygio matavimas [0], daugiasluoksnių medžiagų sluoksnių storio matavimas [29] ir kt. Robotikoje atstumo nustatymo ultragarsu panaudojimo pavyzdys [30]. Šaltinyje [31] minima kitokių atstumo matavimo ultragarsu pavyzdžių t. y. patalpos priežiūra, objektų identifikacija ir dujų tėkmės matavimas. Šie pavyzdžiai tai tik nedidelė ultragarso technologijų panaudojimo dalis. Visų šių taikymų pagrindas yra ultragarsinio signalo sklidimo trukmės matavimas.

2.7. Atstumo įvertinimo principai

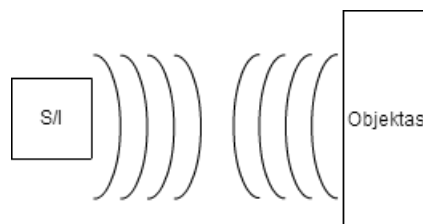
Atstumo įvertinimas ultragarsu pagrįstas ultragarsinio signalo sklidimo trukmės nustatymu (angl. *time-of-flight TOF*) [32]. Sklidimo trukmės atžvilgiu išmatuotas signalo vėlinimas dėl bangos sklidimo terpe (oru) iki objekto, iki kurio matuojamas atstumas, gali būti panaudojamas šiam atstumui įvertinti.

Nuo matavimo sistemos ultragarsinių jutiklių išdėstymo konfigūracijos ir naudojamos signalų apdorojimo įrangos priklauso atstumo matavimui taikoma metodika.

Galimos, tačiau ne visos, jutiklių konfigūracijos pateikiamos 3,4 pav. diagramose:



3 pav. Atstumo matavimas naudojant atskirų siųstuvo ir imtuvo jutiklių sistemą



4 pav. Atstumo matavimas naudojant siųstuvą ir imtuvą suintegruotus į bendrą jutiklį

Trečiame paveiksle pateikiama konfigūracija, kai ultragarsinę atstumo matavimo sistemą sudaro atskiri S-siųstuvas, I-imtuvas. Siųstuvas paskleidžia į aplinką ultragarso bangą. Ši atsispindi nuo objekto ir imtuve gaunamas suvėlintas siųstuvo signalas. Esant šiai konfigūracijai galima panaudoti du garso bangos paskleidimo/priėmimo principus t. y. *pastovus bangų skleidimas* (angl., *continuous wave method-CW*) [33] ir *impulso-aido principas* (angl., *pulse-echo method-PE*) [34, 35]. Pastovus bangų skleidimas/priėmimas vykdomas nuolat generuojant siųstuvą žadinančius impulsus. [36] šaltinio teigimu, CW principas naudojamas kartu su *fazės postūmio metodu*, kuriuo atstumo matavimas pagrįstas skaičiuojant bangų kiekį vykstant fazės postūmiui. PE principu pradžioje siųstuvas, sužadintas DSP sistemos impulsų sekos, į aplinką paskleidžia ultragarso bangas. Šios atsispindėjusios nuo objekto grįžta į imtuvą, kurio signalas apdorojamas DSP sistemos paskaičiuojant TOF. Ketvirtame paveiksle skirtingai nuo 3 pav. siųstuvas ir imtuvas kombinuojami į vieną bendrą jutiklį. Tokioje konfigūracijoje galima taikyti tik PE principą, kuris leidžia realizuoti paprastą, mažos kainos atstumo matavimo sprendimą [37].

2.8. Ultragarso bangos sklidimo trukmės nustatymo metodų analizė

Skyrelyje nagrinėjama klasikinė metodika naudojama sprendžiant pagrindinę ultragarsu paremto atstumo matavimo problemą: ultragarso bangos sklidimo trukmės nustatymą. Literatūroje teigiama, kad vienas iš geriausių ir paprasčiausių būdų ultragarso bangos sklidimo trukmei pamatuoti yra koreliacinio metodo taikymas. Šio metodo teikiamo rezultato maksimali reikšmė išduoda vėlinimo laiką pagal kurį galima paskaičiuoti atstumą.

2.8.1. Koreliacinis metodas

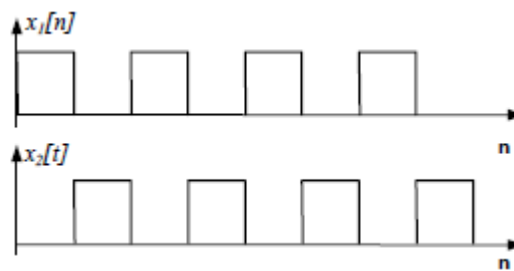
Koreliaciniai metodai yra labai paplitę radiolokacijos ir ultragarsinės regos sistemose. Šie metodai aprašyti [38, 39]. Koreliacija r_{12} leidžia kiekybiškai įvertinti dviejų diskrečių atsitiktinių dydžių $x_1[n]$ ir $x_2[n]$ tarpusavio priklausomybę:

$$r_{12} = \frac{1}{N} \sum_{n=0}^{N-1} x_1[n] \cdot x_2[n]. \quad (1)$$

Kai $x_1[t]$ ir $x_2[t]$ yra susiję tiesiogine priklausomybe r_{12} įgyja maksimalią reikšmę, o kai $x_1[n]$ ir $x_2[n]$ yra nepriklausomi atsitiktiniai dydžiai $r_{12} = 0$.

Anksčiau pateiktas koreliacijos apibrėžimas nėra tinkamas naudoti praktikoje, nes egzistuoja atvejis, kai pagal šį apibrėžimą koreliacija bus lygi 0, nors signalai $x_1[n]$ ir $x_2[n]$ bus 100% koreliuoti.

Toks pavyzdys galėtų būti priešingų fazių signalo koreliacija:



5 pav. Priešingų fazių signalai

Kai $x_1=1$, visada $x_2=0$ ir atvirkščiai, todėl taikant formulę, rezultatas visada bus nulių suma. Todėl norint išvengti šios problemos, signalą x_2 reikia pastumti per j atskaitų į kairę, kad išlyginti signalus. Tokiu atveju gaunama daug koreliacijos reikšmių prie skirtingų postūmio j reikšmių ir dėl to gaunama koreliacijos funkcija. Jeigu $x_1[n]$ pažymimas iš ultragarsinės sistemos išsiųstas signalas $x[n]$, o $x_2[n]$ priimtas atspindžio signalas $yp[n]$, tai gaunama koreliacinės funkcijos diskretinė išraiška:

$$r[j] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot yp[n + j], \quad j = -(N-1), \dots, N-1. \quad (2)$$

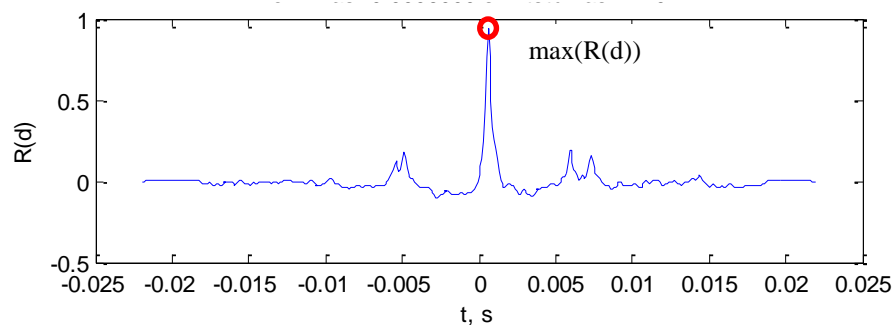
Tuomet postūmio dydis j , prie kurio koreliacinė funkcija įgyja maksimalią reikšmę, atitiks vėlinimo laiką τ , kurį ultragarso banga užtrunka sklisti iki priešais jutiklį esantį objektą ir nuo jo atsispindėti:

$$\tau = \arg \max_j r[j] = \arg \max_j \sum_{n=0}^{N-1} x[n] \cdot yp[n + j]. \quad (3)$$

Ši išraiška yra lengvai realizuojama bet kurioje skaičiavimo architektūroje, todėl bus pritaikyta atstumo matavimo sistemos matematinio modelio realizacijoje.

2.8.2. Maksimumo paieškos metodas

Maksimumo paieškai atlikti galima panaudoti standartinį būdą kuomet peržvelgiamos visos koreliacijos rezultato $R(d)$ (ankstesniame skyrelyje $r(j)$) reikšmės iš jų išrenkant didžiausią. Rezultato iliustracija pateikiama žemiau:



6 pav. Maksimumo paieškos rezultato iliustracija

Naudojant tradicinę DSP signalų apdorojimo įrangą registruojamas signalas yra su triukšmais. Dėl to, šio metodo taikymas tiesiogiai yra neefektyvus, tačiau šiame darbe priimama, kad ultragarso signalas registruojamas ultragarsui apdoroti pritaikyta aparatine įranga, t. y. DSP procesorius su signalo filtravimu nuo triukšmų, todėl šis metodas yra tinkamas maksimumo paieškai.

2.9. Atstumo paskaičiavimas pagal koreliacinės funkcijos rezultatą

Įvertinus koreliacinės funkcijos maksimalią reikšmę ir paskaičiavus ultragarso bangos sklidimo vėlinimą τ (3), sekančiai paskaičiuojamas atstumas D :

$$D = \frac{c \cdot \tau}{2} \quad (4)$$

Čia:

τ – ultragarso bangos atspindžio nuo objekto vėlinimo laikas, s

c – garso greitis (ore) jūros lygyje 340,29 m/s

Formulėje dalyba iš dviejų naudojama dvigubam ultragarso bangos sklidimo keliui įvertinti ir eliminuoti atsiradusią atstumo skaičiavimo paklaidą.

Ši atstumo skaičiavimo formulė yra pagrindas skaičiuojant atstumą ultragarsu. Tai patvirtina šaltinis [32].

2.10. Analizės išvados

1. Atlikta analizė, kurios tikslas nustatyti, kokios šiuo metu yra panašios ir analogiškos sistemos į planuojamą kurti sistemą. Nustatyta, kad tokios sistemos, kaip „Google Toyota Prius“, kuris gali manevruoti gatvėmis be vairuotojo pagalbos, dar greitu laiku nepasirodys, kaip teigia patys Google atstovai. Kaina taip pat nėra skelbiama, tačiau bent šiuo metu ji, manoma, būtų aukšta dėl brangios bei gausios įrangos, reikalingos aplinkos tyrimui ir skaičiavimams.
2. Vertinant analogiškas sistemas, tokias kaip Volvo „Miesto saugumo“ (*City Safety*), Ford „Aktyvus miesto stabdymas“ (*Active City Stop*), išsiaiškinta, jog atstumo nustatymui iki kliūties buvo naudojama „LIDAR“ technologija (lazeriu paremta atstumo aptikimo technologija) ir jog aptinkamas atstumas iki kliūties yra apie 7 metrus.
3. Kuriamai sistemai buvo parinktas ultragarsu paremtas atstumo aptikimas iki kliūties, nes aptikimo atstumas yra toks pat kaip ir analogiškų sistemų, kurios naudoja lazerį, tačiau kainos atžvilgiu, ultragarsinė įranga yra kur kas pigesnė. Taigi kuriama sistema veiks lygiai taip pat, tačiau vartotojui bus galimybė tokią įrangą įsigyti už priimtinesnę kainą, taipogi analogiškų sistemų nėra galimybės įsigyti atskirai, šią sistemą galima pasirinkti perkant kaip papildomą priedėlį prie automobilio.
4. Atlikus analizę, nerasta, kokių būdu analogiškos sistemos skaičiuoja stabdymo atstumą, yra daroma prielaida, jog jie gali naudoti duomenų bazę su stabdymo reikšmėmis, kurios nustatytos bandymais. Tokiai prielaidai pagrindo duoda ir tai, jog tokios sistemos nėra galimybės įsidėti į kiekvieną jų siūlomą automobilio modelį, o tik į kai kuriuos.
5. Išanalizavus keletą komunikacinių protokolų, kuriuos automobiliai naudoja su įdiegtais ECU kuriamai sistemai pasirinktas yra CAN. Šis bus pritaikomas kaip komunikacinis protokolas reikiamiems duomenims iš ECU išgauti. Iš čia kyla reikalavimas automobiliui. Jis turi palaikyti CAN protokolo standartą. Žinoma, jog šį standartą jau nuo 2008 metų naudoja visi transporto priemonių gamintojai. Šaltinio [25] teigimu, daugelis automobilių gamintojų

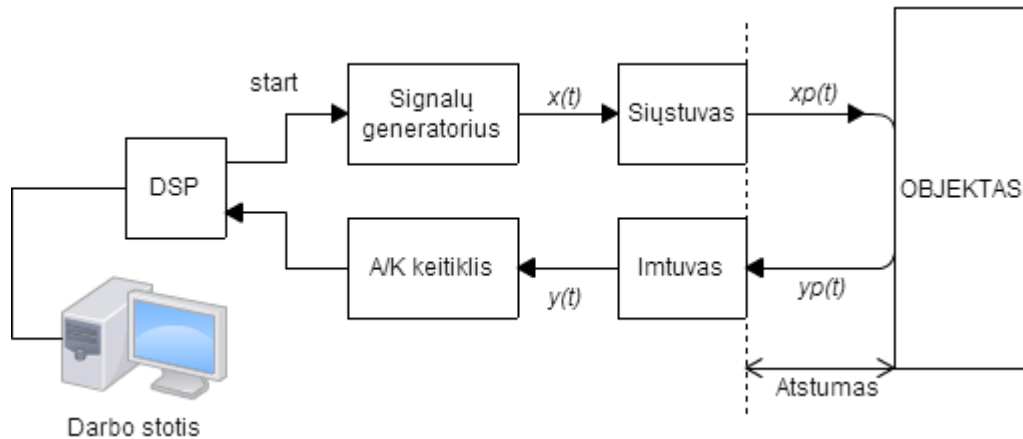
- naudoja jų nuosavus duomenų perdavimo standartus arba perduoda savos konfigūracijos duomenis per CAN duomenų magistralę.
6. Analizuojant ultragarsu paremtą atstumo matavimo metodiką pasirenkama realizuoti atstumo matavimą vienu jutikliu, kurį galima pagrįsti mažesne sistemos realizavimo kaina ir atliekamais paprastesniais skaičiavimais.
 7. Kuriant sistemą tikslinga naudoti CAN protokolą ir sudaryti nuosavus duomenis taip, kad nebūtų pakenkiama normaliam automobilio sistemos darbui.

3. Atstumo matavimo kompiuterinės sistemos modelis

Šiame skyrelyje pateiktas siūlomas sistemos teorinis modelis. Pateikiamas atstumo skaičiavimo algoritmas ir jo patikrinimas MATLAB aplinkoje.

3.1. Atstumo matavimo sistema

Atstumo matavimo sistemą sudarantys komponentai pateikti 7 pav. diagramoje.



7 pav. Atstumo matavimo sistemos struktūra

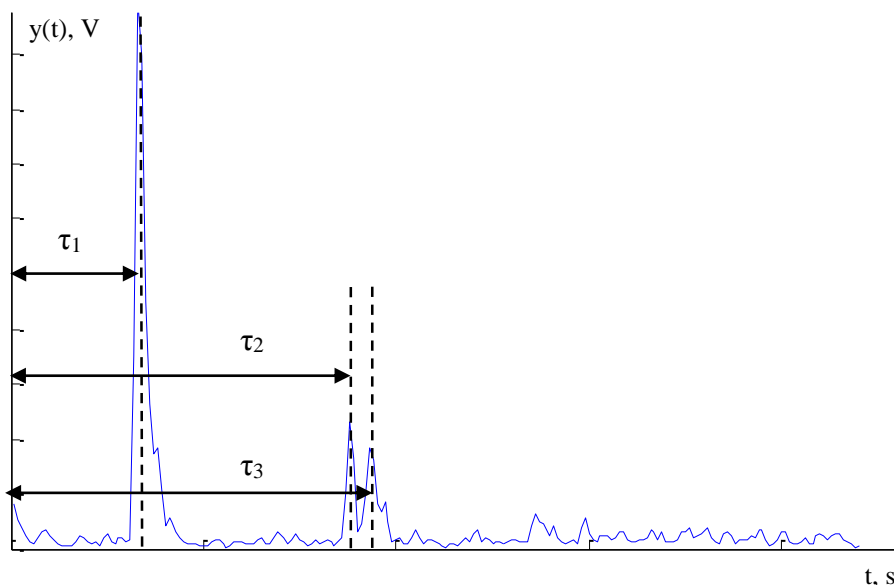
Skaitmeninių signalų procesoriaus (DSP) modulio pagrindinė paskirtis yra valdyti informacijos apdorojimo procesą ir atlikti skaičiavimus su skaitmeniniais signalais. Kiekvieno matavimų ciklo pradžioje DSP generuoja START signalą. Signalų generatorius generuoja signalą $x(t)$ ir siunčia jį ultragarsinį siųstuvą, kuris paskleidžia ultragarso bangą į aplinką. 7 pav. priimama, jog sistemos jutiklių veikimo aplinka yra oras, kuris gali būti su kietų dalelių priemaišomis, pavyzdžiui, purvas ir pan. Tokioje aplinkoje ultragarso bangos yra mažai absorbuojamos ir turi mažą akustinę varžą.

Paskleistas signalas $x_p(t)$ pasiekia objekto, iki kurio matuojamas atstumas, paviršių. Dalis šio signalo prasiskverbia į objektą, tačiau didžioji dalis atsispindi nuo objekto paviršiaus ir grįžta į ultragarso imtuvą $y_p(t)$ signalo pavidalu praėjus laikui Δt nuo išspinduliavimo pradžios.

7 pav. diagramoje pateikiama situacija kuomet atstumas idealiu atveju matuojamas iki vieno objekto, dėl ko būtų gaunamas vienas atspindys, tačiau realiomis sąlygomis atstumas matuojamas aplinkoje, kurioje yra daug objektų. Dėl to, registruojamas signalas $y_p(t)$ imtuve gali būti laikomas kaip visų atspindžių nuo objektų suma:

$$y(t) = \sum_i y_{p_i}(t) \quad (5)$$

Šio signalo pavyzdys pateikiamas sekančiai:



8 pav. Registruojamo signalo $y(t)$ pavyzdys

Imtūve gautas signalas $y(t)$ yra perduodamas į analogas/kodas (A/K) keitiklį, kur jis diskretizuojamas periodu Δt_s ir skaitmeninis signalas perduodamas į signalų apdorojimo procesorių (DSP). DSP atlieka skaičiavimus pagal vėliau pateikiamą algoritmą. Atliekamų skaičiavimų rezultatas yra 7 pav. pažymėtas „Atstumas“. Šį atstumą pažymint simboliu D atstumas paskaičiuojamas pagal formulę (4) apibrėžtą skyrelyje 2.9.

3.2. Koreliacijos metodo pritaikymas atstumo paskaičiavimui

Darbe atstumo skaičiavimams atlikti panaudojamas klasikinis koreliacijos metodas. Metodus panaudojamas vėlinimams laike tarp dviejų signalų x ir y nustatyti. Koreliacijos rezultato maksimali reikšmė parodo, kiek signalas y yra pasislinkęs laike x signalo atžvilgiu. Skirtingai nuo koreliacijos apibrėžimo (2) darbe realizuojamo koreliacijos metodo matematinė išraiška pateikiama formule:

$$R(d) = \frac{\sum_i [(x(i) - mx) \cdot (y(i + d) - my)]}{\sqrt{\sum_i (x(i) - mx)^2} \cdot \sqrt{\sum_i (y(i) - my)^2}} \quad (6)$$

Čia:

$R(d)$ – koreliacijos metodo rezultatas esant vėlinimui d , kur $d \in [-(N-1), \dots, (N-1)]$

N – koreliuojamų sekų ilgiai

mx, my – atitinkamai x, y koreliuojamųjų sekų vidurkiai

i – žingsnis per vėlinimus d

Formulės (6) vardiklyje įvertinama abiejų sekų x, y dispersija.

Skaičiuojant koreliaciją priimama, kad x tai skaitmeninis ultragarso signalas vadinamas *etaloniniu signalu* x_{ref} , kuris pamatuojamas žinomu atstumu (pavyzdžiui 1 metras, 8 pav.). Tokiai prielaidai esant, atstumo skaičiavimo formulė (4) pakeičiama į sekančią formulę:

$$D = \frac{c \cdot \tau_i}{2} + atstumas(x_{ref}) \quad (7)$$

3.3. Matematinis modelis MATLAB aplinkoje

Ketvirtoje lentelėje pateikiama atstumo matavimo sistemos matematinis modelis, kuriame išskiriamos matavimų ir skaičiavimų fazės.

4 lentelė. Atstumo matavimo sistemos matematinis modelis

Matavimų fazė
1. Išmatuojami signalai x_{ref} , y .
Skaičiavimų fazė
1. $R = koreliacija(x_{ref}, y)$
2. $\tau(R) = \arg \max_d R$
3. $D = \frac{c \cdot \tau(R)}{2} + atstumas(x_{ref})$

Sistemos matavimų fazėje registruojami signalai:

- x_{ref} – pamatuotas signalas žinomu atstumu iki objekto. Matavimų metu signalas fiksuojamas esant 1 m atstumui. Šis signalas registruojamas tik vieną kartą.
- y – signalas, registruotas esant *neetaloniniam* atstumui.

Skaičiavimų fazės veiksmai 1-3 kartojami kiekvieno kartojimo pradžioje fiksuojant vis naują y signalą.

Pirmame punkte paskaičiuojama koreliacija tarp etaloninio ir neetaloninio signalo.

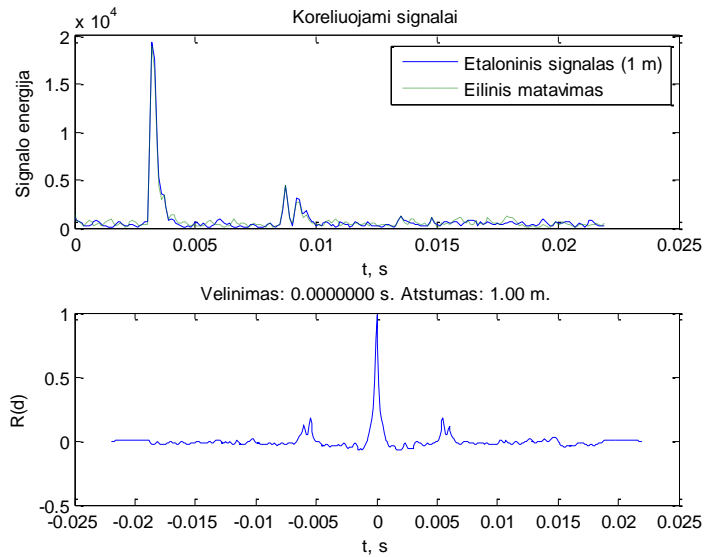
Antrame punkte randama koreliacijos rezultato R maksimali vertė pagal kurią paskaičiuojamas vėlinimas laiko ašyje.

Trečiame punkte paskaičiuojamas atstumas įvertinant etaloninio signalo atstumą ir neetaloninio signalo vėlinimo laiką.

Matematinio modelio realizacijos programinis kodas pateiktas A priede. Esminės atstumo skaičiavimo detalės pateiktos pajuodintai.

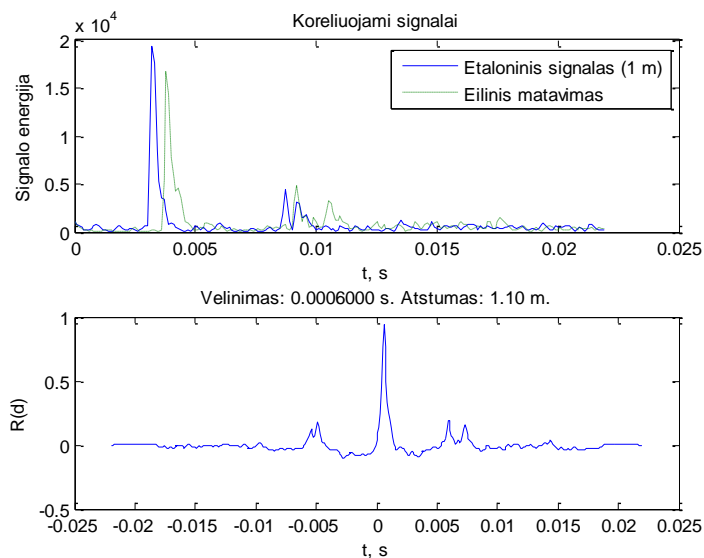
3.4. Matematinio modelio patikrinimas

Realizuoto matematinio modelio skaičiavimai atliekami MATLAB aplinkoje. Patikrinimui naudojami ultragarso signalo duomenys pamatuoti *Maxq7667 evaluation kit* plokšte. Sekančiuose grafikuose pateikiami skaičiavimų rezultatai kuomet matuojama 1 ir 1.1 m atstumais:



9 pav. Matavimai esant 1 m atstumui

Matyti, kad 9 pav. koreliuojamas etaloninis signalas su eilinio matavimo signalu. Šio signalo energija visoje laiko ašyje t sutampa su etaloniniu signalu, todėl atliktą koreliaciją $R(d)$ galima laikyti *etaloninio signalo autokoreliacija*, kai signalas koreliuojamas pats su savimi. Remiantis koreliacijos rezultatu pagal (4) formulę paskaičiuotas atstumas turėtų būti lygus 0, tačiau įvertinus etaloninio signalo atstumą (7) formulėje skaičiuojamas atstumas yra 1 m.



10 pav. Matavimai esant 1.1 m atstumui

Matuojant 1.1 m atstumą 10 pav. matomas ryškus eilinio signalo atspindžio poslinkis laiko ašyje, kas iššaukia koreliacijos rezultato $R(d)$ maksimumo poslinkį teigiama vėlinimų ašies kryptimi.

Paskaičiuotas vėlinimas $\tau(R)$ lygus $600 \mu\text{s}$ dėl kurio skaičiuojamas atstumas padidėja 10-čia centimetrų lyginant su etaloninio signalo atstumu.

Toliau darbe pateikiami naudojamoms DSP platformoms apibrėžti matavimų ir skaičiavimų algoritmai.

3.5. Atstumo nustatymo matavimų fazės algoritmas

Skyrelyje pateikiamas matavimų algoritmo aprašas.

Algoritmo derinimo parametrai:

Parametrai nustatyti eksperimentiškai tinkantys registruoti signalą, kuriuo galima paskaičiuoti atstumą iki 4 metrų.

$imp = 5$ – impulsų kiekis pakankamas ultragarsiniam 40 kHz dažnio jutikliui sužadinti,

$T_s = 100 \mu\text{s}$ – ultragarso signalo diskretizavimo periodas, dažnis 10 kHz,

$SignaloAtsKiekis = 250$ – pakankamas signalo atskaitų kiekis išmatuoti iki 4 metrų atstumą.

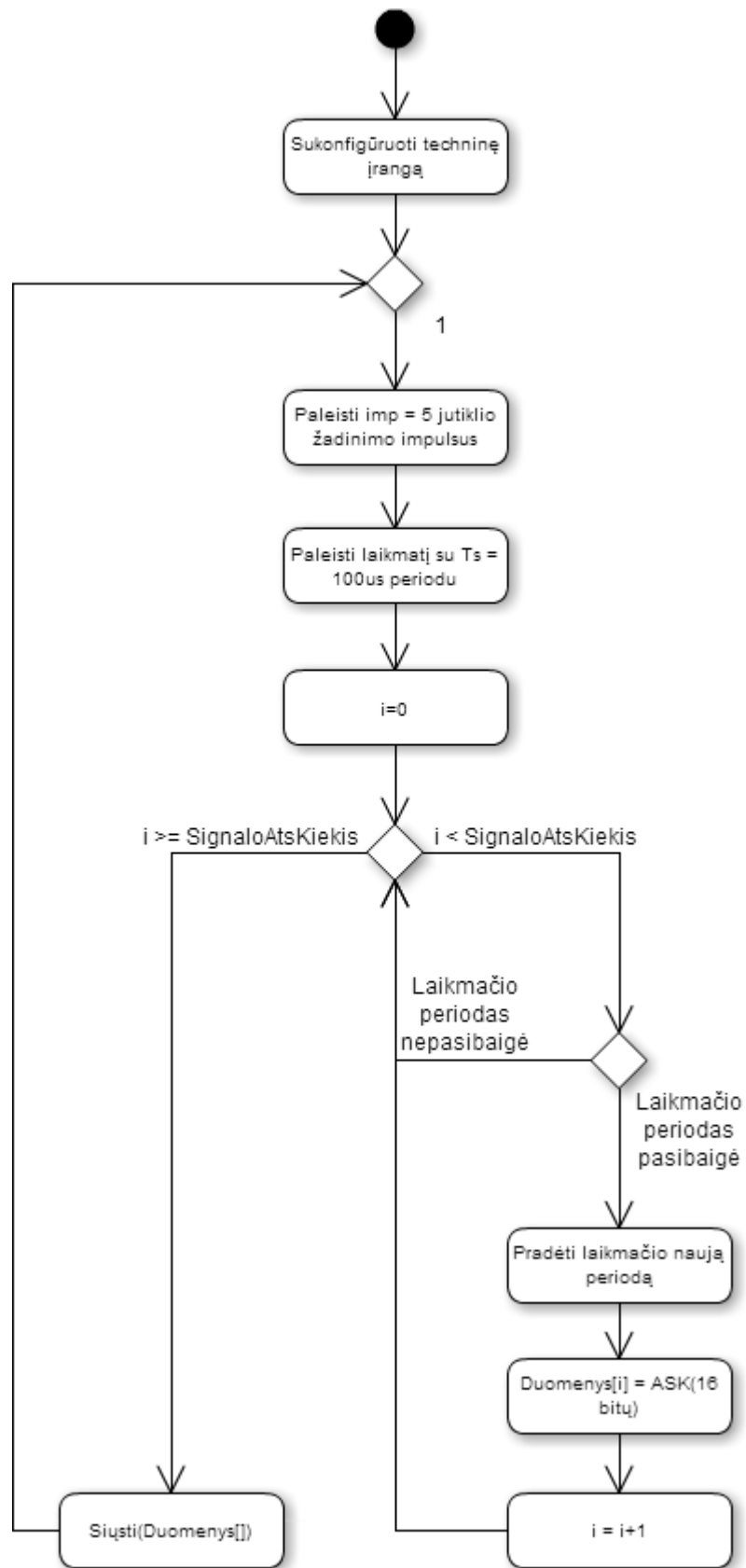
Rezultatai:

Užregistruoti ultragarso signalai:

$x_{ref}[250]$ – etaloninis signalas su žinomu atstumu (1 m),

$y_{echo}[250]$ – signalas, kurio atstumas $< 4 \text{ m}$.

Algoritmo UML veiklos diagrama pateikta 11 paveiksle.



11 pav. Ultragarso signalo registravimo modulio veikimo algoritmas

3.6. Koreliacijos skaičiavimo algoritmas

Šiame skyriuje pateikiamas pilno koreliacijos algoritmo aprašas, kurio vidiniai koreliacinės sumos kaupimo ir maksimumo fiksavimo algoritmai detalizuojami sekančiuose skyreliuose.

Pradiniai duomenys:

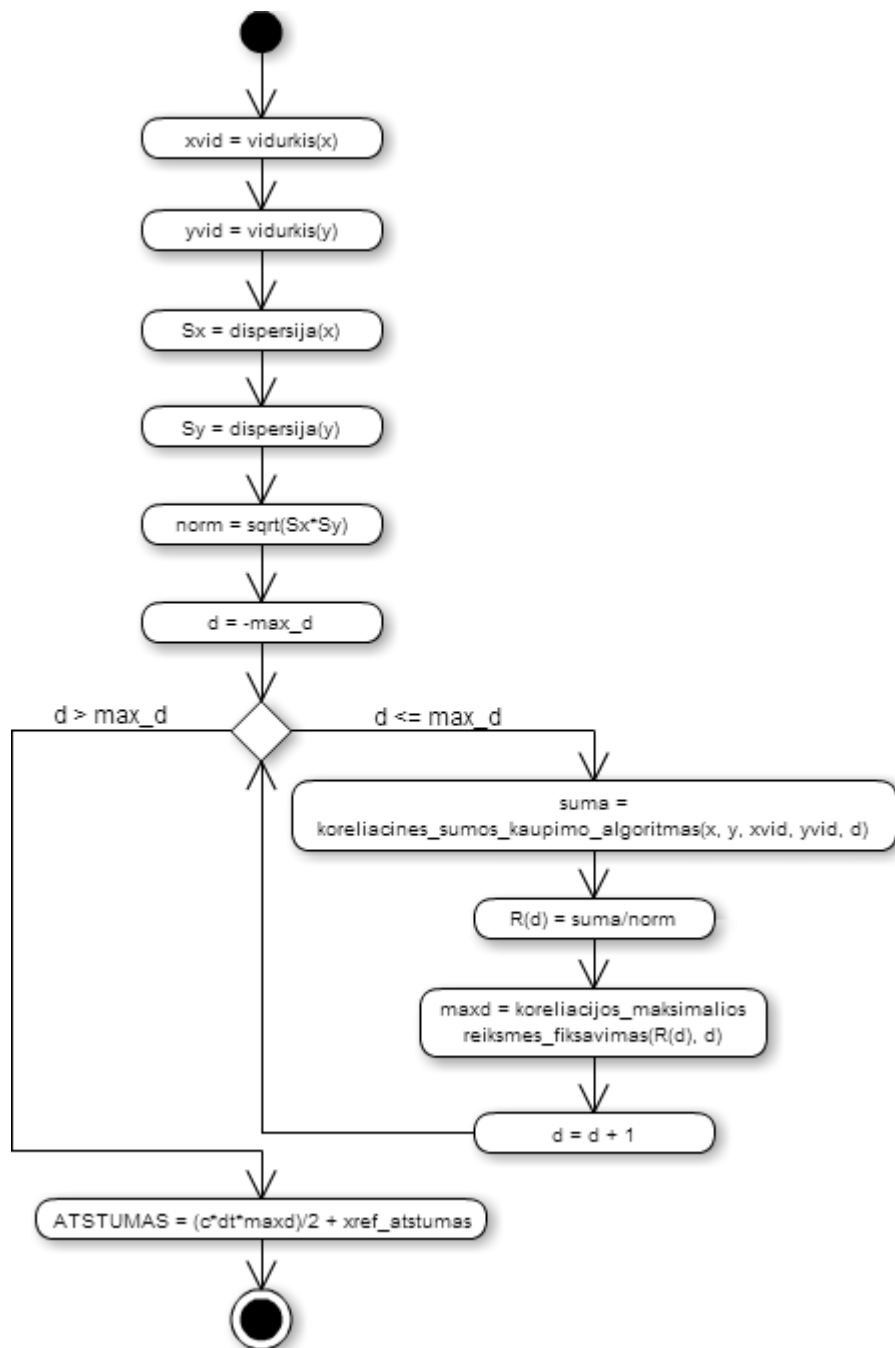
$x(1..xlen)$ – registruotas signalas x_{ref} (žr. skyr. 3.5),
 $y(1..ylen)$ – registruotas signalas y_{echo} (žr. skyr. 3.5),
 $xlen = ylen = N = 220$ – koreliuojamų sekų ilgiai,
 $max_d = -(N-1) = -219$ – koreliacijos skaičiavimo pradinis vėlinimas.

Skaičiavimų parametrai:

x_{vid} , y_{vid} , S_x , S_y – įvertinami koreliuojamų sekų vidurkiai, dispersijos, kad būtų galima paskaičiuoti normalizavimo koeficientą $norm$,
 $norm$ – šiuo koeficientu normalizuojamos koreliacijos rezultato reikšmės.

Rezultatai:

$R(1..(2*N)-1)$ – sekų x , y koreliavimo rezultatas,
 $ATSTUMAS$ – pagal gautą koreliacinės funkcijos rezultatą paskaičiuotas atstumas metrais.



12 pav. Pagrindinis koreliacijos algoritmas

3.7. Koreliacinės sumos kaupimo algoritmas

Šiame algoritme atliekama koreliacinės sumos kaupimas įvertinant koreliuojamų sekų vidurkius.

Pradiniai duomenys:

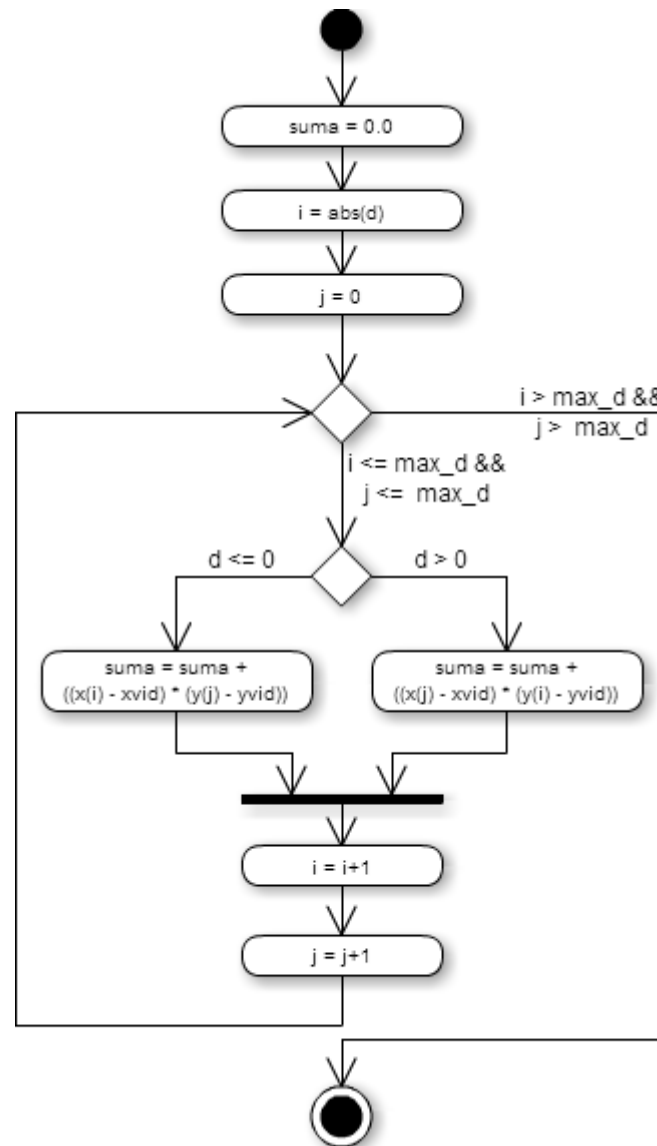
x, y – koreliuojamos sekos aprašytos 3.6 skyrelyje,

x_{vid}, y_{vid} – atitinkamai x, y sekų vidurkiai,

d – vėlinimas, kuriam skaičiuojama suma.

Rezultatai:

$suma$ – sumos kaupimo rezultatas.



13 pav. Koreliacinės sumos kaupimo algoritmas

3.8. Koreliacijos maksimalios reikšmės fiksavimo algoritmas

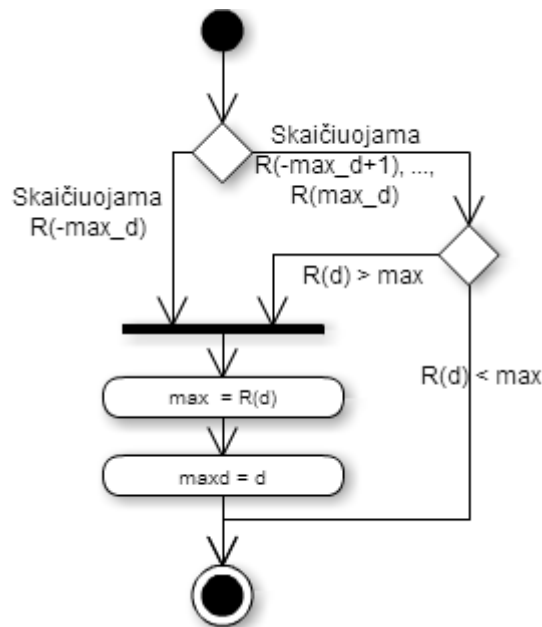
Algoritmas realizuojamas kaip pagrindinio koreliacijos metodo dalis. Iteraciniu būdu skaičiuojant $R(d)$ reikšmę tikrinama ar yra dar didesnė $R(d)$ reikšmė, nei yra užfiksuota max reikšmėje.

Pradiniai duomenys:

$R(d)$ – koreliacijos rezultatas, kur $d = [-N+1, \dots, N-1]$.

Rezultatas:

$maxd$ – vėlinimas d , pagal kurį koreliacijos rezultatas įgyja maksimalią reikšmę.



14 pav. Koreliacijos maksimalios reikšmės fiksavimo algoritmas

3.9. Matematinio modelio sudarymo išvados

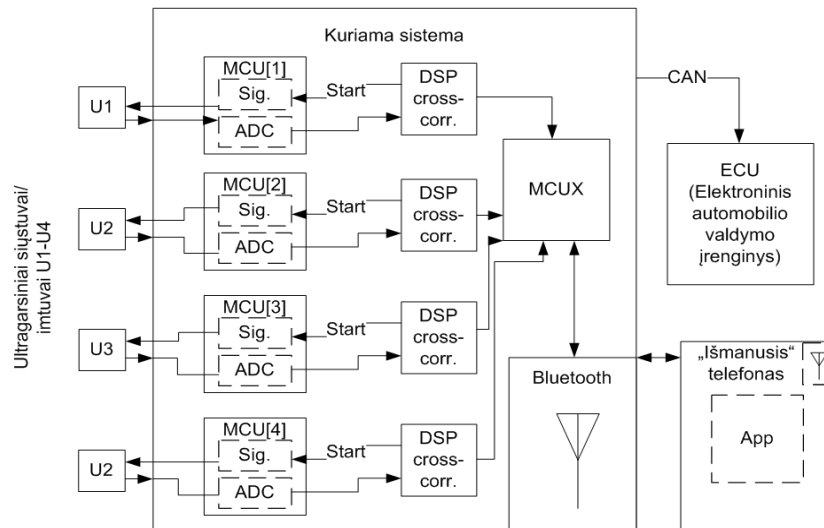
1. Atliekant matematinio modelio sudarymą pateikta principinė modelio schema į kurią diegiamas vėliau pateiktas matematinis modelis.
2. Modelio pagrindą sudaro koreliacinio metodo taikymas, kuris literatūros teigimu yra pats paprasčiausias būdas atstumo skaičiavimams atlikti.
3. Matematinis modelis realizuotas MATLAB aplinkoje ir patikrintas atliekamų skaičiavimų teisingumas. Rezultatas pateiktas grafikais.
4. Pateikti matematinį modelį sudarančių metodų algoritmai UML diagramomis.

Žinoma, jog koreliacinis metodas yra imlus skaičiavimo resursams. Pateiktame koreliacijos algoritme įvertinama koreliuojamų sekų vidurkiai, dispersijos ir atliekama daug sumos kaupimo operacijų, kurių kiekį su koreliuojamų sekų ilgiu sieja kvadratinė priklausomybė. Vidurkiai ir dispersijos naudojami koreliacijos rezultatui normalizuoti. Šie skaičiavimai labai prailgina atliekamų skaičiavimų trukmę. Sekančio skyriaus eksperimentinėje dalyje pateiktas šių skaičiavimų trukmės įvertinimas konkrečioje DSP platformoje ir palyginama kaip pagerinamas vykdymo laikas, kai koreliacijos skaičiavimai sumažinami iki koreliacinės sumos, maksimumo radimo ir galiausiai atstumo paskaičiavimo.

4. Praktinė matematinio modelio realizacija

4.1. Sistemos koncepcijos sudarymas

Pateikiama sudaryta kompiuterinės sistemos koncepcija (15 pav.), kurioje atsižvelgiama į sistemos sąsają su automobilyje jau esančia kompiuterine įranga ir ryšį su sistemos vartotoju.



15 pav. Siūlomos sistemos struktūrinė schema

Siūlomos sistemos koncepcinį modelį sudaro trys pagrindinės matavimų sistemos dalys. 15 pav. pateiktoje schemoje ją sudaro ultragarsiniai jutikliai (U1..U4), jutiklių signalus registruojantys valdikliai (MCU1,..., MCU4). Diskretizuoti jutiklių matavimų duomenys patenka į DSP (angl. *digital signal processor*). Čia pritaikomi sistemos veikimo pagrindiniam uždaviniui (matuoti atstumą) spręsti naudotini algoritmai. Kuriamoje sistemoje pasirinktas standartiškai, atstumui matuoti taikomas, koreliacijos algoritmas.

Sistema diegiama į automobilį, todėl reikalinga sistemos dalis (MCUX), palaikanti ryšį su automobilyje jau esančiomis elektroninėmis sistemomis, kurios garsu ar vaizdu galėtų įspėti vairuotoją apie pavojingą atstumą iki kliūtis. Palaikant ryšį su automobilio ECU įrenginiu yra galimybė papildyti kuriamos sistemos funkcionalumą arba padidintų atliekamų matavimų tikslumą įvertinant automobilio judėjimo greitį ar kitų automobilyje įdiegtų jutiklių duomenis, pvz., temperatūros jutiklis, padangų sukibimo su kelio danga jutikliai ir pan.

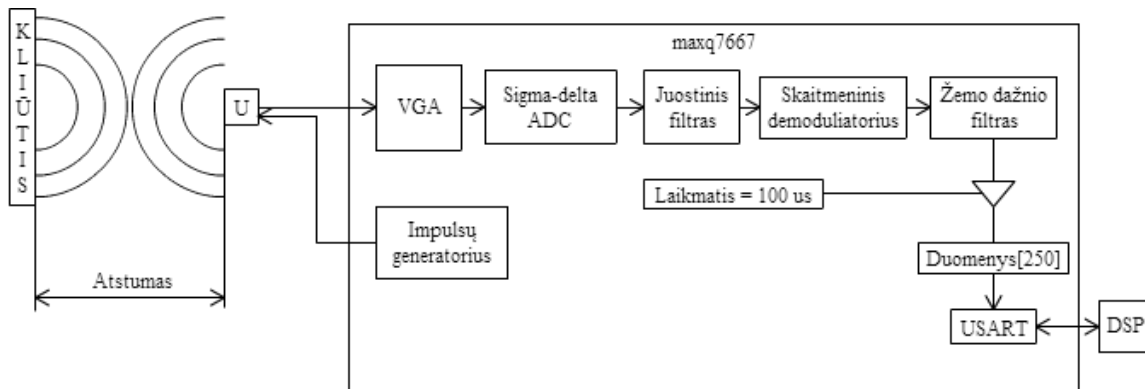
Bet kokios naujų technologijų sistemos palankiai vertinamos savybės:

- Sistemos išmanumas – kai sistemos veikimui reikalaujama minimalus vartotojo veiksmų kiekis;
- Sistemos dydis – kai sistema telpa vartotojo delne. Tokį dydį leidžia realizuoti šiuolaikinės lustinės technologijos.

Įvertinant apibrėžtas savybes, sistemos koncepcijoje panaudotinas bevielės ryšys su išmaniaisiais telefonais, taip pat sistemai suteikiant galimybę, kad bet kuriuo metu būtų galima atjungti nuo automobilio elektroninės sistemos, taip papildomai apsaugant automobilio bateriją nuo energijos išsekimo.

4.2. Ultragarso signalo registravimo techninė įranga

Pagal koncepciją (15 pav.), sistemoje išsiskiria matavimų ir skaičiavimų posistemės. Darbo sistemoje naudojamo ultragarsinio jutiklio rezonansinis dažnis 40 KHz.



16 pav. Ultragarso signalą registruojantis sistemos procesorius

4.2.1. Matavimų posistemė ir jos veikimas

Darbo realizacijoje matavimams atlikti panaudojama plokštė su joje specialiai ultragarso matavimams skirtu procesoriumi maxq7667. Procesoriaus techniniai parametrai ir naudojami vidiniai techniniai komponentai:

- Darbinis dažnis 16 MHz;
- Impulsų sekos generatorius – generuoja impulsų seką, kurios dažnis lygus naudojamam ultragarsinio jutiklio rezonansiniam dažniui;
- VGA (angl. *Variable Gain Amplifier*) kintamasis stiprintuvas; stiprina 49dB su 7.21 mVp-p amplitude;
- Pasiekiamą skaitmeninės duomenų reikšmės rezoliucija 0.11 μ Vp-p/LSB;
- 16 bitų sigma-delta ADC (Analogas-kodas keitiklis);
- Skaitmeniniai filtrai (juostinis ir žemo dažnio);
- Nuoseklus duomenų perdavimas (USART).

Pagal 4 pav. schemą matavimai atliekami pagal žemiau pateikiamus veiksmus:

- Ultragarsinis jutiklis sužadinas impulsų generatoriumi generuojant 40 KHz dažnio impulsus, taip paskleidžiant ultragarso bangą į aplinką;
- Priklausomai nuo to kokie objektai yra jutiklio regėjimo lauke, ir koku atstumu, nuo jų atsispindi jutikliui būdingo dažnio garsas su triukšmais ir grįžta į jutiklį;
- Jutiklyje gautas garsas dėl jame esančio pjezokristalo elemento paverčiamas į elektrinį signalą;
- Jutiklio signalas silpnas, todėl procesoriuje esantis stiprintuvas (VGA) signalą sustiprina pagal anksčiau pateiktus parametrus;
- Sustiprintas elektrinis signalas skaitmenizuojamas 16 bitų sigma-delta ADC (analogas-kodas keitikliu);
- Skaitmeninis juostinis filtras seka generuojamo signalo dažnį skaitmenizuotame signale;
- Skaitmeninis demoduliatorius ir žemo dažnio filtras sukuria signalo apvaskalą (angl. *echo envelope*) su 16 bitų rezoliucija. Šis apvaskalas apdorojamas taikant skaitmeninio signalo apdorojimo metodus ir algoritmus.

4.3. Eksperimentinio tyrimo metodika

Šių eksperimentų tikslas yra įdiegti atstumo matavimo algoritmą į realią skaičiavimo architektūrą ir atlikti jo spartos ir tikslumo matavimus lyginant darbe taikomo koreliacijos metodo realizacijas naudojant ir nenaudojant specifinių DSP procesoriaus operacijų bei koreliacijai reikalingų skaičiavimų. Darbo metu Matlab aplinkoje sukurto matematinio modelio realizacija įrašoma į konkrečią DSP architektūrą. Realizacija atliekama ANSI C kalba. Programos kodas pateiktas B priede.

4.3.1. Eksperimento priemonės

4.3.1.1. dsPIC platforma

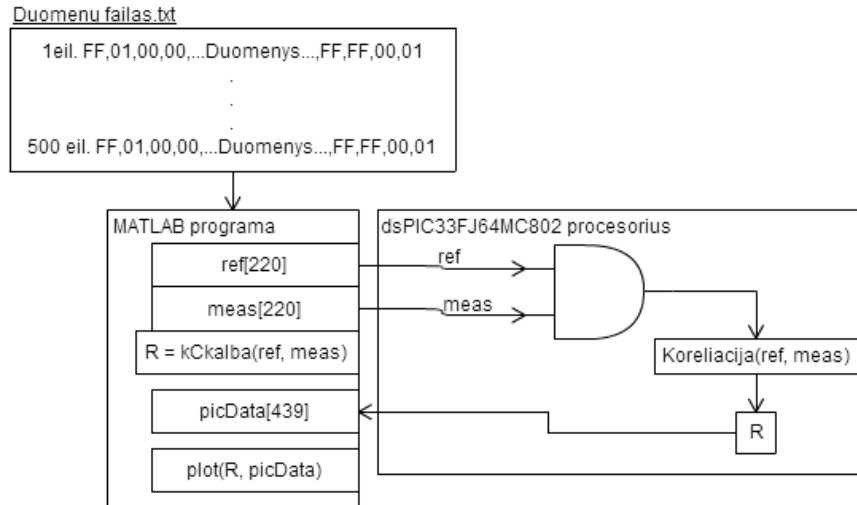
Eksperimentams atlikti ir praktiškai pamatuoti darbe vystyto algoritmo vykdymo trukmes panaudota Microchip kompanijos vystymo platforma (angl., *Microstick for dsPIC33F and PIC24H devices*) su joje įdiegtu dsPIC33F serijos procesoriumi, kurio duomenys pateikti žemiau:

5 lentelė. Darbe naudojamos DSP platformos parametrai

Procesorius:	dsPIC33FJ64MC802
Taktinis dažnis:	80 MHz, 40 MIPS
Vidinė magistralė:	16-bitų
Adresuojama atmintis:	64 KB
Atmintinė:	16 KB SDRAM duomenims, 64 KB Flash programos atmintis
Procesoriaus architektūra:	16-bitų modifikuota Harvard architektūra
Kompiliatorius:	MPLAB XC16-GCC (v1.21), DSP biblioteka
Kompiliatoriaus vėliavėlės:	-O0, -fast-math, -fno-short-double, -mlarge-code -mlarge-data -menable-large-arrays -falign-arrays -funroll-loops

4.3.2. Eksperimento vykdymo schema

Pateikta 17 pav. schema, kurią sudaro tokios pagrindinės dalys: Duomenų_failas.txt, Matlab aplinkos programa ir DSP procesorius.



17 pav. Eksperimento vykdymo schema

Duomenų_failas.txt faile saugojami duomenys, kurie surenkami registruojant ultragarso signalą naudojant *maxq7667 evaluation kit* platformą.

Matlab programa atlieka duomenų failo nuskaitymą, duomenų siuntimą į DSP procesorių, kad šis paskaičiuotų koreliaciją ir skaičiavimo rezultatą pasiųstų į Matlab aplinką, kurioje būtų galima atlikti koreliacijos skaičiavimo, matlab ir DSP aplinkose, palyginimą.

DSP procesoriuje realizuojamas šiame darbe apibrėžtas atstumo matavimo matematinis modelis, su kuriuo atliekami eksperimentiniai tyrimai. Sudarytosios (17 pav.) schemas MATLAB realizacija pateikta C priede.

4.3.3. Spartos matavimo metodika

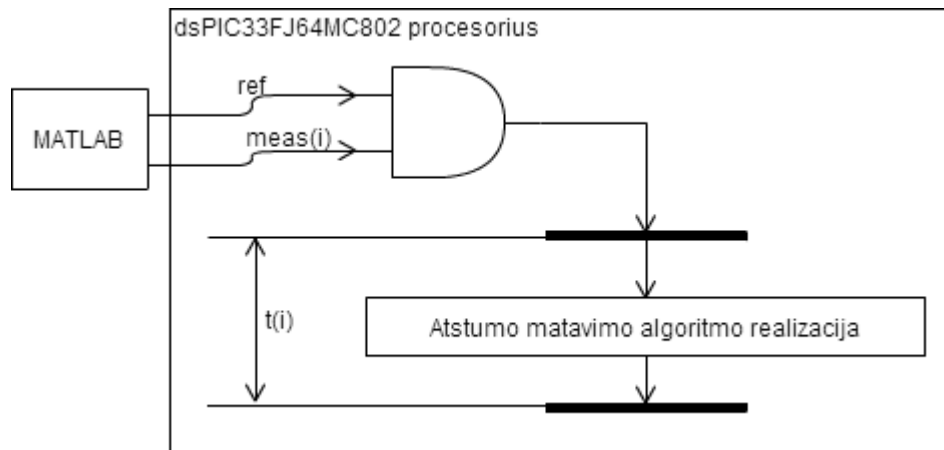
Algoritmų realizacijos vykdymo spartą galima matuoti dviem būdais:

- skaičiuoti procesoriaus ciklus,
- matuoti algoritmo vykdymo trukmę.

Pirmuoju būdu reikia skaičiuoti procesoriaus ciklų kiekį vykdant algoritmo realizaciją. Toks matavimo metodas yra labai tikslus. Tačiau jo esminis trūkumas yra tai, kad skirtingose eksperimentinėse platformose reikia naudoti skirtingas ciklų matavimo priemones.

Pasirinktas kombinuotas būdas, kuomet naudojant eksperimentinės platformos procesoriaus laikmatį paskaičiuojamas procesoriaus ciklų kiekis, kuris vėliau panaudojamas algoritmo vykdymo trukmei paskaičiuoti įvertinant procesoriaus instrukcijų vykdymo dažnį. Metodas yra paprastas ir jam realizuoti pakanka standartinių mikroprocesorinių priemonių. Pagrindinis tokio metodo trūkumas yra tas, kad matavimo tikslumas priklauso nuo naudojamų kompiliatorių ir pačios aparatinės įrangos architektūros. Darbe naudojama eksperimentų vykdymo platforma gali matuoti laiką tokiu tikslumu kokį suteikia paskaičiuota vykdymo trukmė naudojant slankaus taško, XC16 kompiliatoriaus tipą *long double*. Pasiekiamas iki 8 skaičių po kablelio tikslumas. Tokio matavimo tikslumo pakanka ilgų ir trumpų procesų trukmei matuoti, tačiau konkrečiu atveju atliekant sistemos matematinio modelio algoritmo trukmės skaičiavimus pastebėta, kad taikomo koreliacinio metodo vykdymo trukmė priklauso nuo koreliuojamų signalų formos. Ši problema yra sprendžiama matuojamą algoritmą vykdant M kartų ir skaičiuojant bendrą vykdymo laiką t ir jo vidurkį (matavimo schema pateikta 18 pav.).

Atstumo matematinis modelis diegiamas į DSP procesorių, kuriame realizuojama skaičiavimų spartos matavimo schema (žr. pav.):



18 pav. Algoritmo vykdymo trukmės matavimo schema

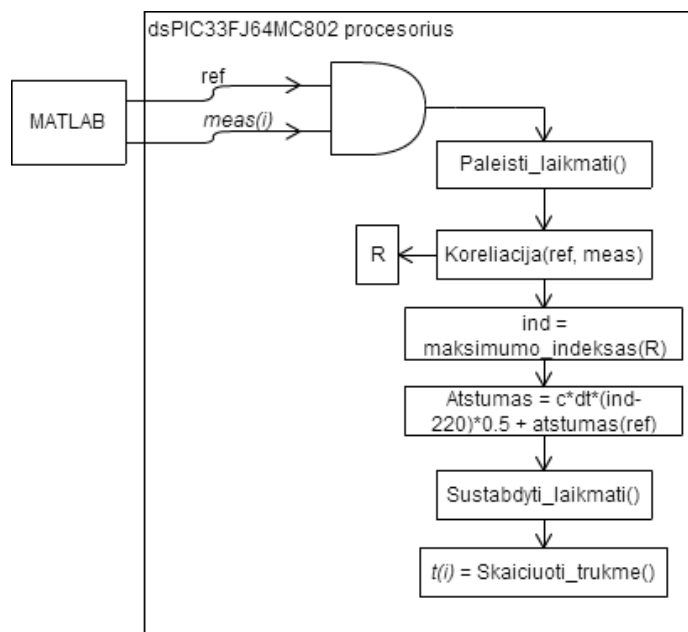
Algoritmo vykdymo trukmių reikšmės $t(i)$ paskaičiuojamos siunčiant į procesorių vis naują $meas(i)$ matavimo signalą, kurio atžvilgiu toliau įvykdoma atstumo matavimo algoritmo realizacija, kuriai įvykus fiksuojama trukmė. Kaskart gauta i -toji vykdymo trukmė $t(i)$ saugojama masyve t_{mas} , kurį galutinai užpildžius įvertinamas jo reikšmių vidurkis t_{vid} . Žemiau pateiktas vykdymo spartos įvertinimo algoritmas:

1. $M = 0$
2. $t_{mas}(M) = 0$
3. Paleisti `laikmatį()`
4. Įvykdyti matuojamo algoritmo realizaciją 1 kartą
5. $t_{mas}(M) = sustabdyti_laikmatį()$
6. $M = M + 1$
7. Kartoti 2-6 punktus K kartų
8. Paskaičiuoti t_{vid} pagal sekančią formulę:

$$t_{vid} = \frac{1}{M} \sum_{i=0}^{M-1} t_{mas}(i) \quad (8)$$

Šeštame punkte naudojamo parametro K dydis priklauso nuo to, kiek eilučių yra 17 pav. pateiktame `Duomenų_failas.txt` faile. Jame pateikti ultragarso signalo matavimų duomenys. Pirmoji failo eilutė laikoma, kad yra *etaloninio signalo* duomenys, o sekančios – neetaloninio signalo duomenys. Kas kartą gavus neetaloninį signalą kartojamos 2-6 eilutės aukščiau pateikto algoritmo. Galiausiai visų šių matavimų trukmės patikrinamos oscilografu.

Spartos ir atstumo matavimo algoritmų realizacijos galutinis įdiegimas į DSP procesorių pateikiamas sekančioje schemoje:



19 pav. DSP procesoriuje atliekami skaičiavimai

4.4. Atstumo matavimo spartos įvertinimas ir palyginimas

Spartos matavimas atliekamas dviem atvejais:

- kai koreliacijos algoritmas realizuojamas slankaus taško aritmetikos pagrindu naudojant 12 pav. pateikto algoritmo schemą;
- kai koreliacija realizuojama fiksuoto taško aritmetikos pagrindu atsisakant 12 pav. algoritme atliekamų sekų vidurkių ir dispersijų skaičiavimų.

Sekantiai pateikiamas minimų koreliacijos realizacijų palyginimas:

6 lentelė. Skirtingos standartinės koreliacijos realizacijos

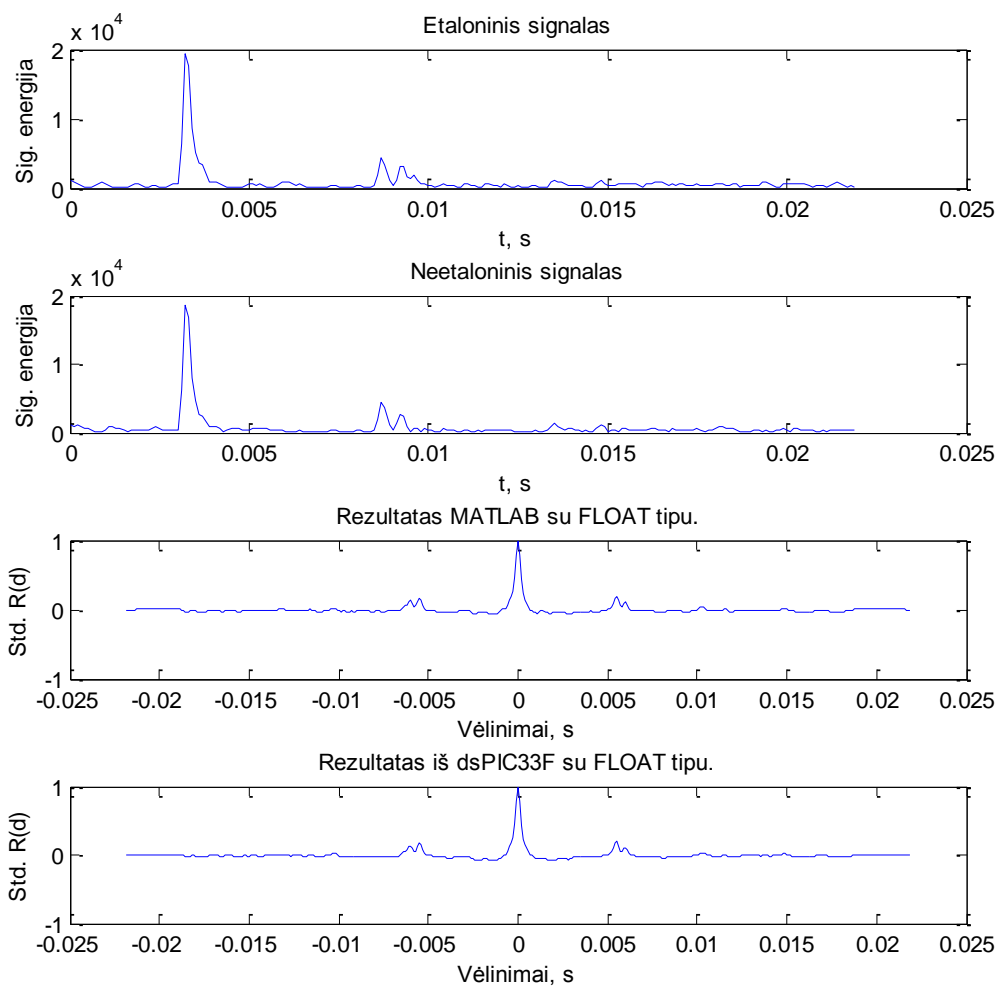
Standartinė koreliacija su slankiu tašku	Standartinė koreliacija su fiksuotu tašku
1. $x_{vid}, y_{vid} = \text{vidurkiai}(x, y)$	1. $\text{maxSuma} = 0$
2. $S_x, S_y = \text{dispersijos}(x, x_{vid}, y, y_{vid})$	2. $d = -\text{max_d}$
3. $\text{normalizavimo_konstanta} = \sqrt{S_x * S_y}$	3. $\text{suma} = \text{koreliacinės_sumos_kaupimas}(x, y, d)$
4. $d = -\text{max_d}$	4. $\text{maxSuma} = \text{koreliacijos_max_reikšmės_fiksavimas}(\text{suma}, d)$
5. $\text{suma} = \text{koreliacinės_sumos_kaupimas}(x, y, x_{vid}, y_{vid}, d)$	5. $d = d + 1$
6. $R(d) = \text{suma} / \text{normalizavimo_konstanta}$	6. Kartoti 2-5 punktus, kai: $d \in [-\text{max_d}, \dots, \text{max_d}]$
7. $\text{maxd} = \text{koreliacijos_max_reikšmės_fiksavimas}(R(d), d)$	7. $\text{ATSTUMAS} = c * dt * \text{maxSuma} * 0.5 + x_{ref_atstumas}$
8. $d = d + 1$	
9. Kartoti 5-8 punktus, kai: $d \in [-\text{max_d}, \dots, \text{max_d}]$	
10. $\text{ATSTUMAS} = c * dt * \text{maxd} * 0.5 + x_{ref_atstumas}$	

Standartinės koreliacijos su fiksuotu tašku pagrindiniai skirtumai nuo slankaus taško koreliacijos:

- nevertinami koreliuojamų sekų vidurkiai ir dispersijos, dėl to neatliekamas rezultato normalizavimas $-1, \dots, 1$ režiuose;
- sumos kaupimas realizuojamas naudojamo DSP procesoriaus assemblerio instrukcijomis taip išnaudojant specifinės DSP operacijos MAC (angl., *multiply-accumulate*) teikiamas galimybes. Šios operacijos metu per vieną instrukcijos vykdymo ciklą atliekama dviejų 16 bitų skaičių daugyba ir sumavimas prie procesoriuje esančio 40 bitų akumulatoriaus;
- standartinė koreliacija su slankiu tašku pilnai realizuota C programavimo kalboje, tuo tarpu fiksuoto taško – C kalba + assembleris.

Pateiktosios realizacijos įrašomos į DSP procesorių, kad būtų galima pamatuoti vykdymo trukmes pagal 4.3.3 skyrelyje pateiktą metodiką. Eksperimentai atliekami, kai koreliuojamų sekų ilgis $N = 220$. Spartos matavimo metodikos parametras $K = 1500$ t. y. tiek kartų paskaičiuojama koreliacija su DSP platforma. Koreliacijos metodų parametras $max_d = 219$, nes pagal sekų ilgį N , $max_d = N-1$. Abiejų koreliacijos realizacijos variantų koreliavimo metu atliekamų sumos operacijų kiekis lygus $N^2 = 48400$.

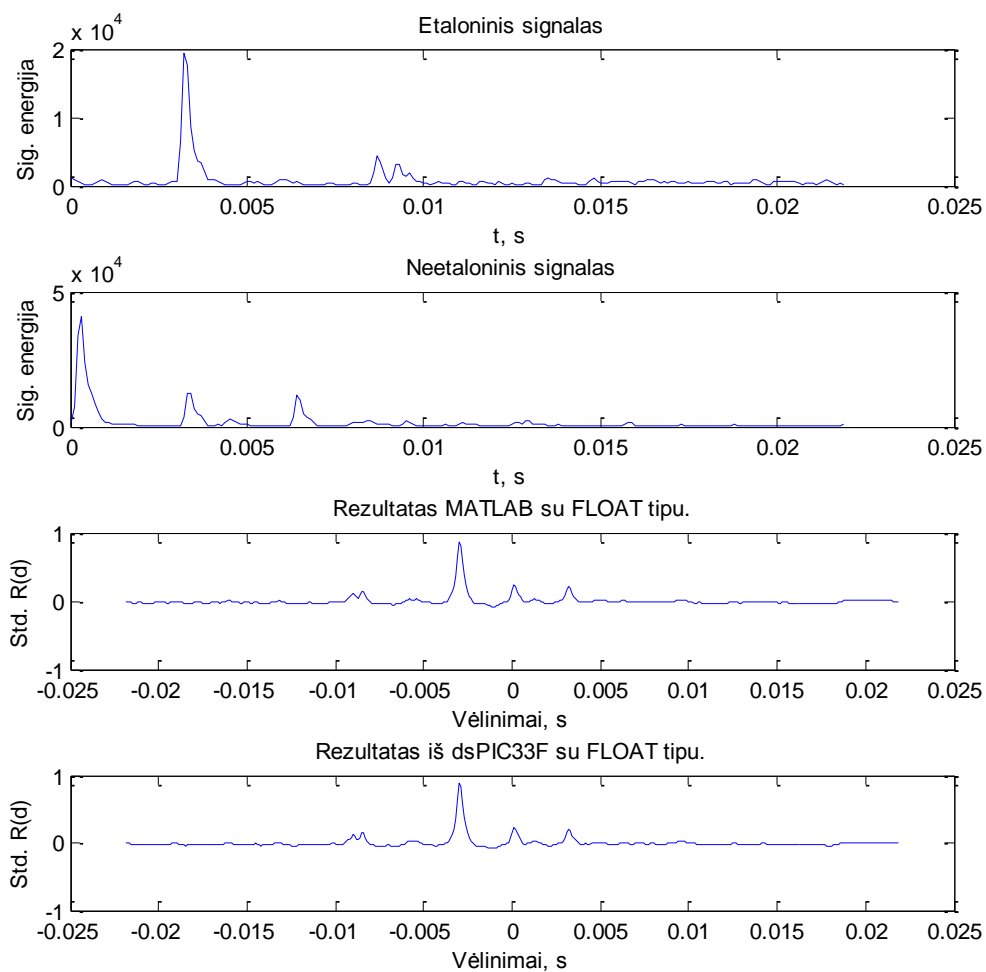
Pradžioje 17 pav. eksperimentų vykdymo schemoje patikrinama ar skaičiavimų rezultatai MATLAB aplinkoje sutampa su rezultatais, kuriuos pateikia dsPIC33F procesorius:



20 pav. 220 ilgio sekų koreliavimo palyginimas kai koreliuojama MATLAB ir dsPIC33F platformoje

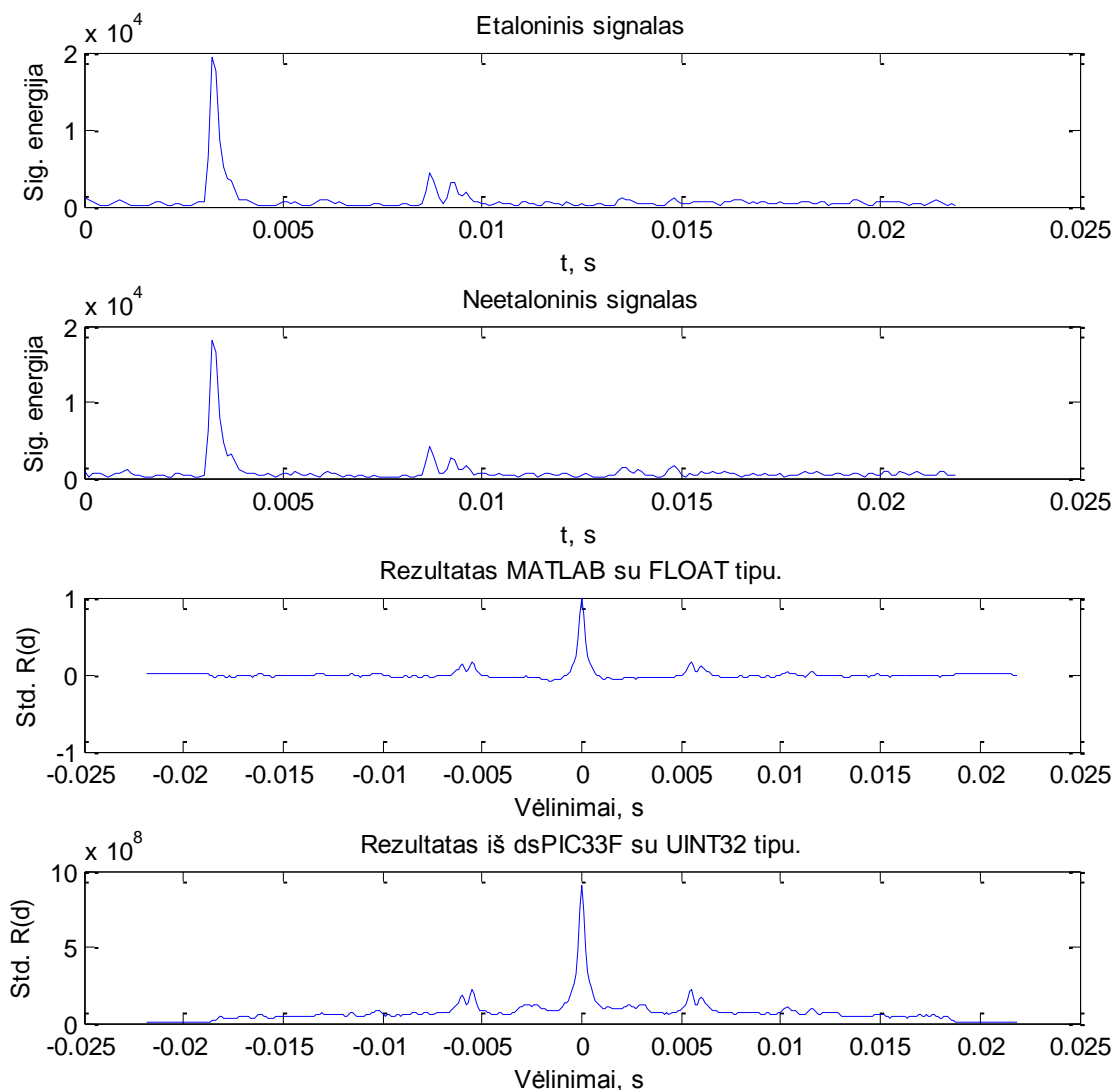
Iš grafiko matyti, kad rezultatai visiškai sutampa, kas patvirtina koreliacijos veikimo dsPIC33F platformoje teisingumą. Rezultatas gautas taikant slankaus taško koreliacijos realizaciją MATLAB ir dsPIC33F procesoriuje.

Sekančiame paveiksle pateikta kaip slankaus taško koreliacija paskaičiuojama esant neigiamam neetaloninio signalo poslinkiui etaloninio signalo atžvilgiu:



21 pav. Koreliacijos skaičiavimai su neetaloninio signalo poslinkiu

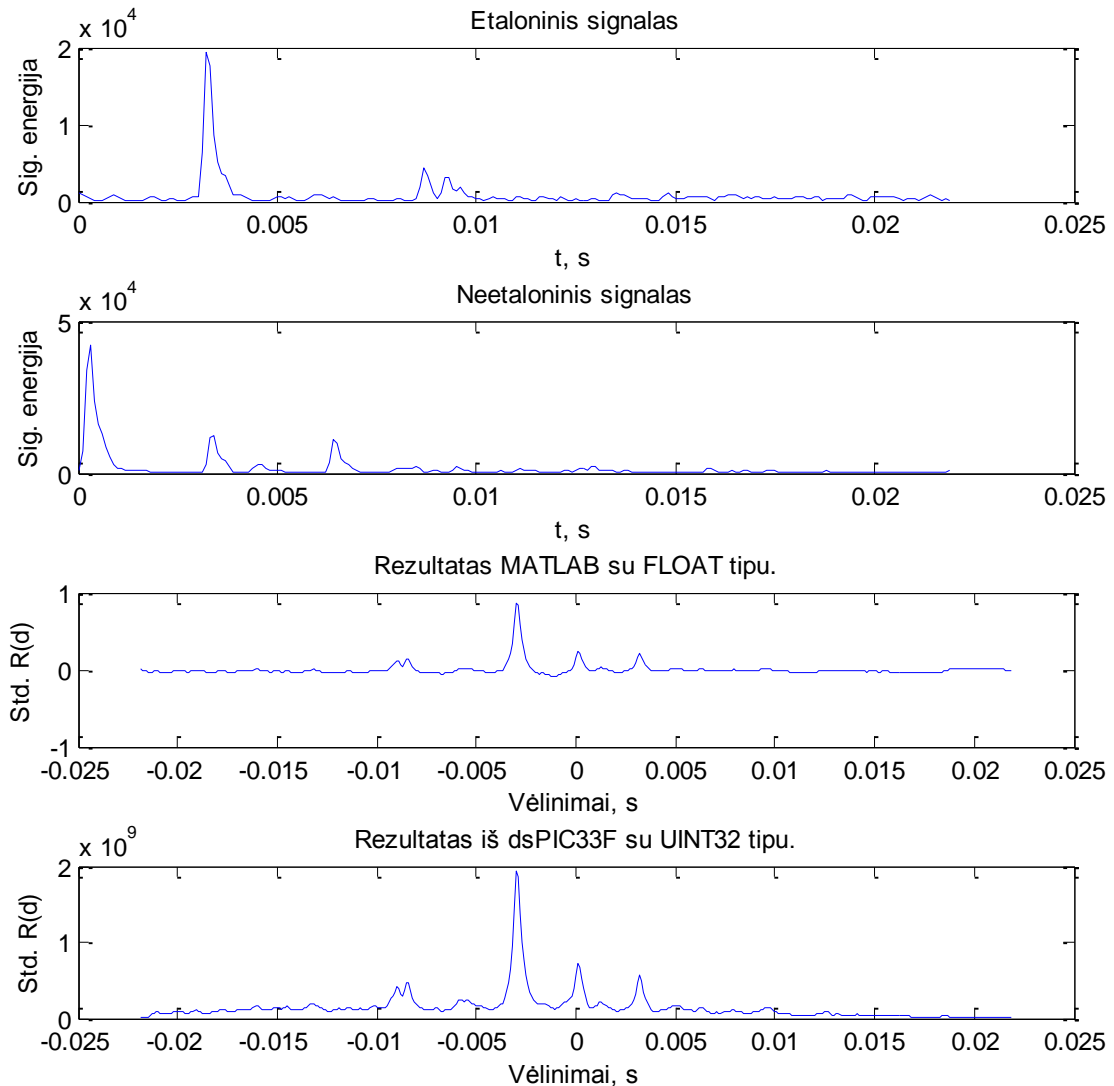
Toliau pateikiami grafikai, kuriuose lyginama fiksuoto ir slankaus taško koreliacijų rezultatai:



22 pav. Fiksuoto ir slankaus taško koreliacijų rezultatų palyginimas

Galima pastebėti, kad 22 pav. DSP koreliacijos rezultatas turi visai kitą reikšmių intervalą. Taip yra dėl to, kad rezultatas nėra normalizuotas. Eksperimentinių skaičiavimų metu nustatyta, kad DSP sumos kaupimo rezultatui pakanka 32 bitų sveikųjų tipo reikšmės, tačiau esant būtinybei galima sumos kaupimą atlikti 64 bitų sveikųjų tipo reikšme, nes DSP procesoriaus sumos kaupimo įrenginys yra 40 bitų pločio.

Sekančiais grafikai pateikiamas rezultatas kuomet koreliacijos skaičiavimai atliekami su neetaloninio signalo poslinkiu.



23 pav. Fiksuoto ir slankaus taško koreliacijų rezultatų palyginimas su poslinkiu

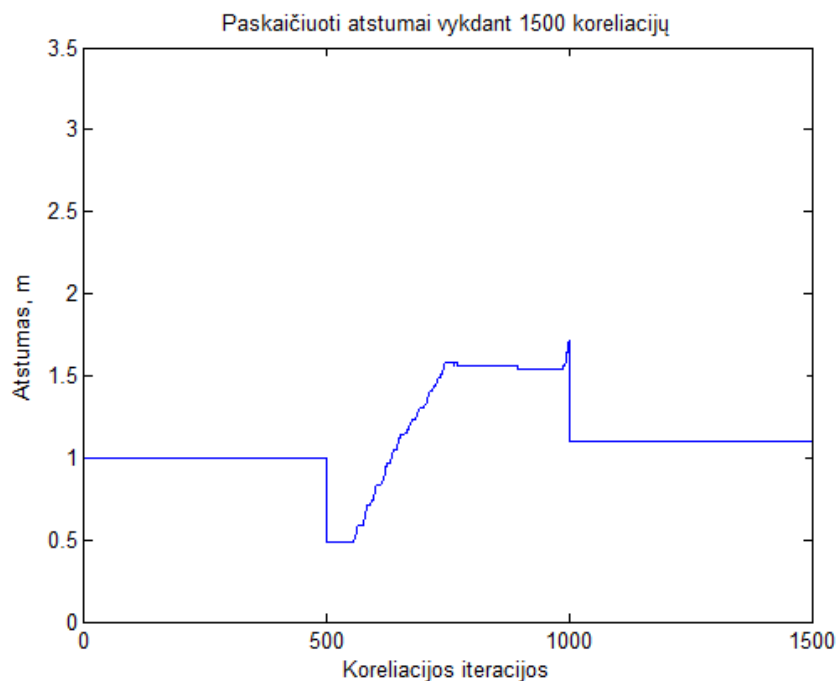
Eksperimentiniame tyrime atliekama 1500 koreliacijos skaičiavimų, kuriems duomenys pateikiami 17 pav. duomenų faile. Kiekvienos koreliacijos rezultate įvertinami paskaičiuoti atstumai ir atstumo matavimo algoritmo vykdymo trukmės. Rezultatai pateikti dviem atvejais:

- kai atstumai ir trukmės paskaičiuojami esant slankaus taško koreliacijos realizacijai;
- kai atstumai ir trukmės paskaičiuojami esant fiksuoto taško koreliacijos realizacijai.

Duomenų faile saugojami tokie duomenys:

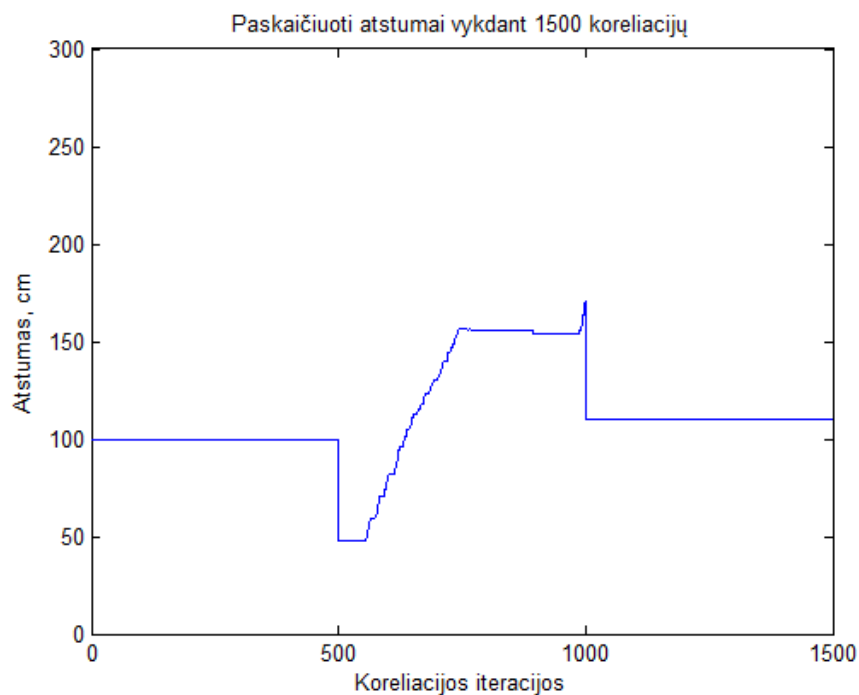
- pirmose (1-500) eilutėse saugojami registruoti ultragarsinio signalo duomenys kuomet matuojamas pastovus 1 metro atstumas;
- kitose (501-1000) eilutėse saugojami registruoti ultragarsinio signalo duomenys kuomet matuojamas kintamas 50cm-1.7 metro atstumas;
- paskutinėse (1001-1500) eilutėse saugojami registruoti ultragarsinio signalo duomenys kuomet matuojamas pastovus 1.1 metro atstumas.

Pateiktas grafikas atliekant skaičiavimus taikant slankaus taško koreliaciją:



24 pav. Slankaus taško koreliacijos pagrindu paskaičiuoti atstumai 1 cm tikslumu

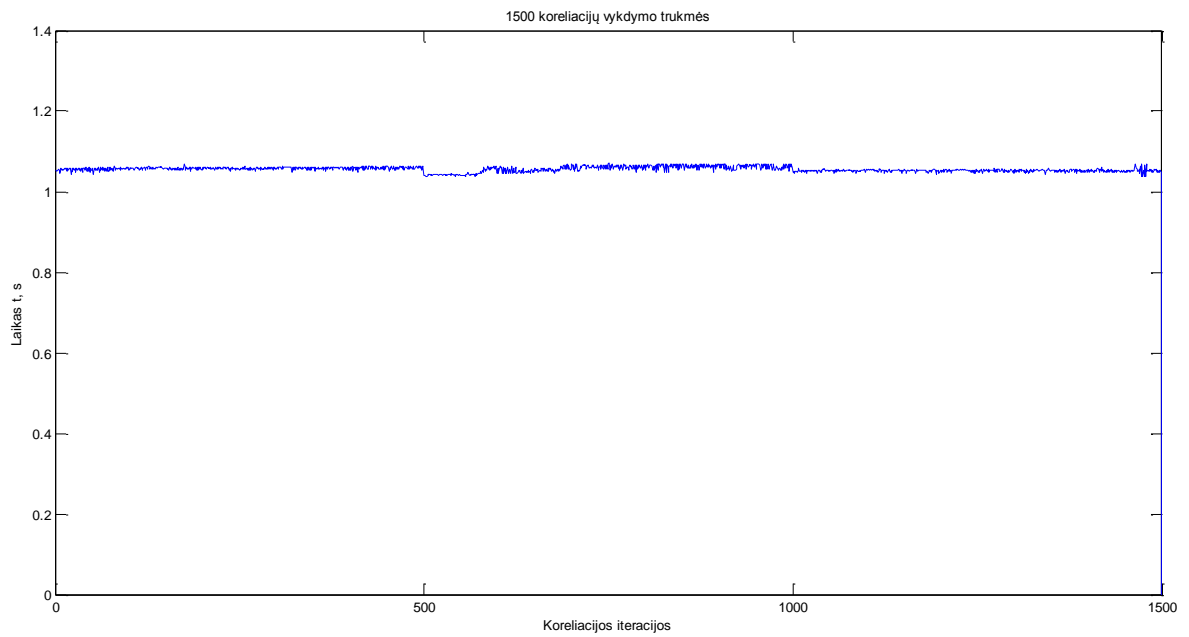
Toliau grafikas atliekant skaičiavimus taikant fiksuoto taško koreliaciją:



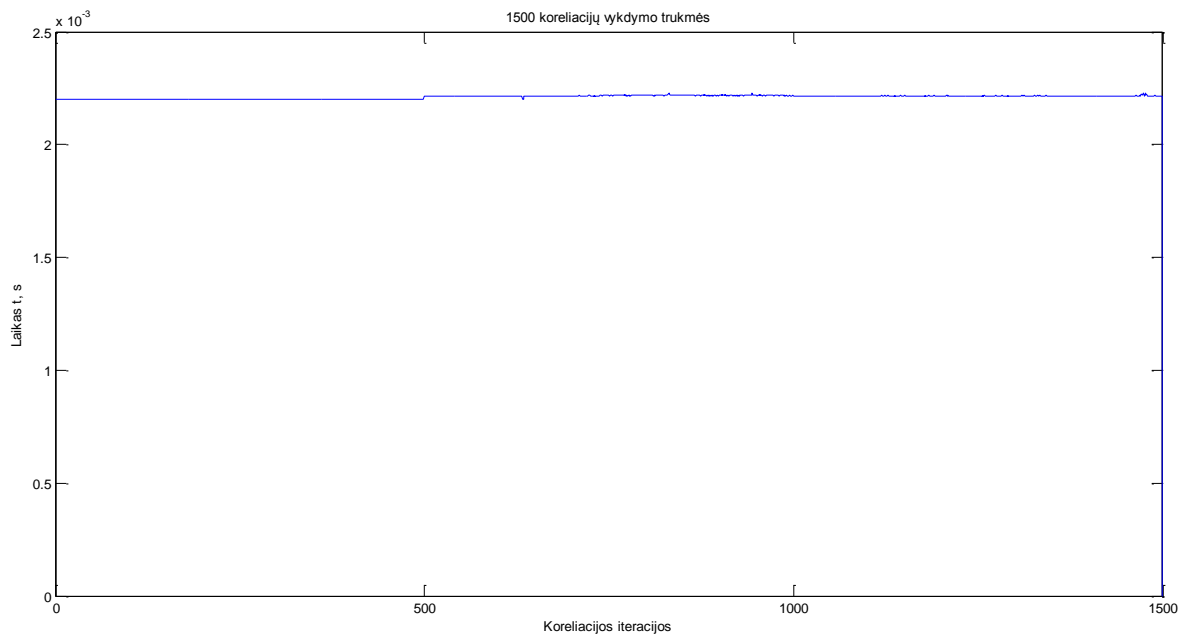
25 pav. Fiksuoto taško koreliacijos pagrindu paskaičiuoti atstumai 1 cm tikslumu

Rezultatas pateiktas išreikštas centimetrais siekiant išvengti slankaus taško skaičiavimų.

Atliktas atstumo matavimo algoritmo vykdymo trukmių nustatymas. Čia analogiškai vykdymo trukmės pamatuojamos taikant slankaus ir fiksuoto taško koreliacijos metodus (26, 27 pav.).



26 pav. Atstumo matavimo vykdymo trukmės slankaus taško koreliacijos atveju



27 pav. Atstumo matavimo vykdymo trukmės fiksuoto taško koreliacijos atveju

Gautuosius vykdymo trukmių rezultatus apibendrina sekančiai pateikta lentelė, kurioje taip pat pateikti oscilografu atliktų matavimų rezultatai:

7 lentelė. Vykdymo trukmių palyginimo lentelė

	Slankaus taško koreliacija	Fiksuoto taško koreliacija	Santykis
Vidurkiai (26, 27 pav. sekų)	1,055 s	0,0022 s	479,5
Oscilografo matavimai	1,06 s	0,0024 s	441,6

Matyti, jog pereinant iš slankaus taško į fiksuoto taško koreliaciją vidutiniškai 460 kartų pagreitėja atliekami skaičiavimai. Tai lemia sumažintas besikartojančių algebrinių operacijų kiekis ir panaudotas procesoriuje esantis DSP variklis, o tiksliau MAC operacija koreliacinėms sumoms skaičiuoti.

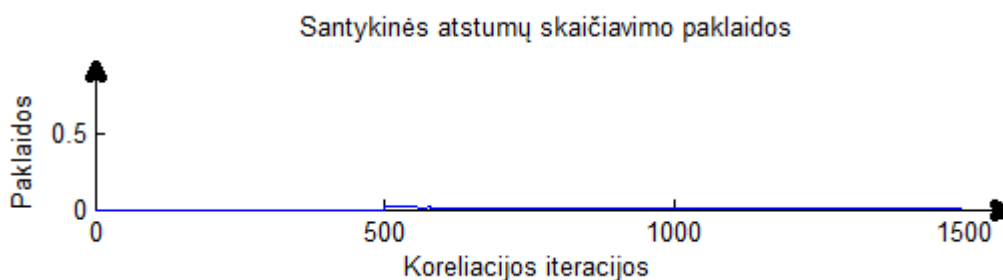
Fiksuoto taško koreliacijos trukmė (2,4 ms), tai tokia trukmė, kai paskaičiavus koreliacijos rezultatą paskaičiuojamas ir atstumas. Atliekamų skaičiavimų dažnis 416 Hz. Maksimalus leistinas greitis Lietuvoje yra 130 km/h autostradoje, todėl automobilis judėdamas pastoviu tokiu greičiu per vieną atstumo paskaičiavimą nuvažiuoja 8,66 cm. Miesto gatvėse, kur galimas 50 km/h greitis atstumas paskaičiuojamas kas 3.3 cm, o 90 km/h regioninės reikšmės keliuose atstumas paskaičiuojamas kas 6 cm.

4.5. Atstumo skaičiavimo tikslumo įvertinimas

Darbe atliekamų atstumo skaičiavimų tikslumas įvertinamas priimant, jog slankaus taško koreliacija paskaičiuoja atstumą tiksliai. Tuo tarpu fiksuoto taško koreliacija paremti atstumo skaičiavimų rezultatai lyginami su atstumais paskaičiuotais naudojant slankų tašką įvertinant santykinę paklaidą, kuri paskaičiuojama pagal sekančią formulę:

$$\text{santykinė_paklaida} = \frac{|x - a|}{a} \quad (9)$$

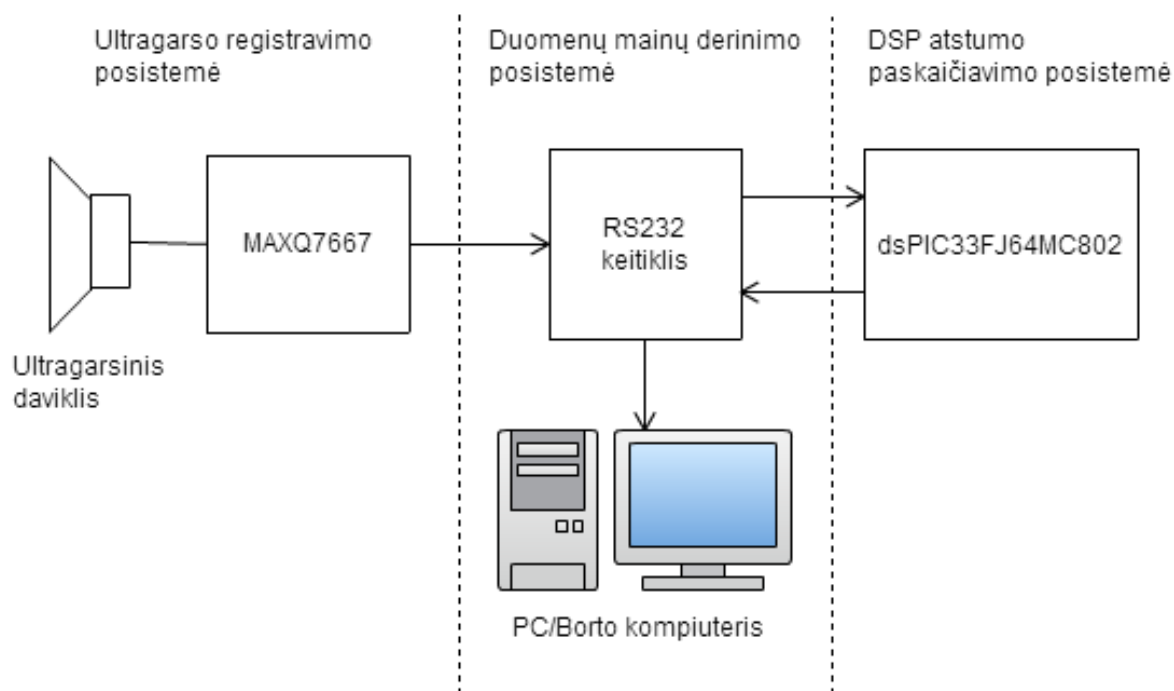
Formulėje x – atstumas, paskaičiuotas taikant slankaus taško koreliaciją, a – fiksuoto taško koreliacijos atstumas. Atstumų skaičiavimo paklaidos pateikiamos sekančiame grafike:



28 pav. Santykinės paklaidos tarp atstumų, paskaičiuotų su slankaus ir fiksuoto taško koreliacijomis

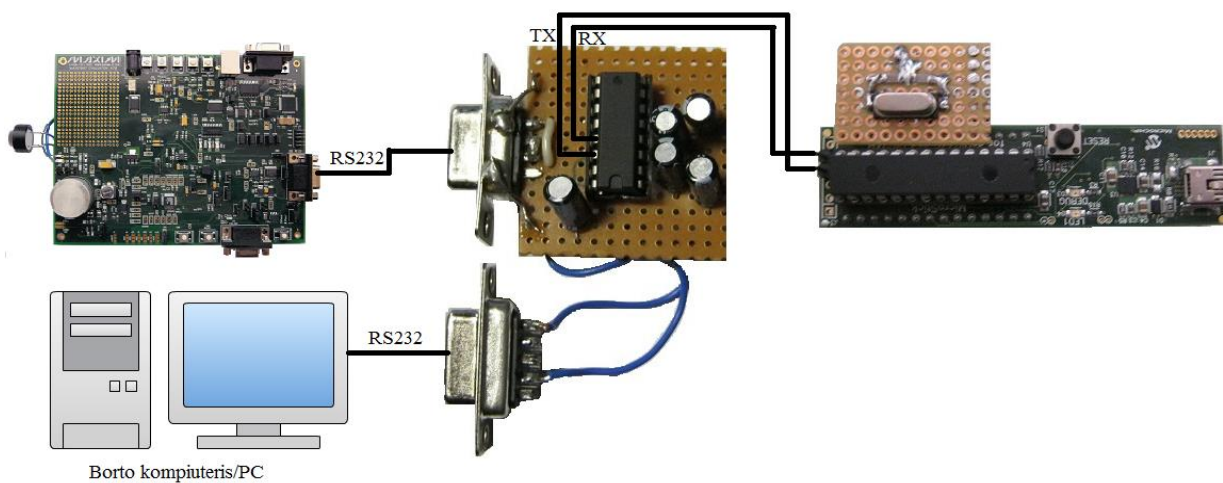
Paveiksle matyti, jog santykinė atstumų skaičiavimo paklaida įgyja maksimalią reikšmę, kai skaičiuojamas atstumas yra mažiausias (50 cm). Šios paklaidos procentinis įvertis yra 1,99%. Esant tokiai paklaidai galima teigti, jog koreliacijos skaičiavimo metodas, realizuotas fiksuoto taško aritmetikos pagrindu yra tinkamas atstumo skaičiavimams atlikti.

Sekančiai pateikta principinė ultragarsinės atstumo matavimo sistemos schema:



29 pav. Ultragarsinės atstumo matavimo sistemos principinė schema

Principinė schema realizuota elektroninėmis schemomis pateikta sekančiai:



30 pav. Ultragarsinė atstumo matavimo sistema

Schemose DSP posistemėje paskaičiuotas atstumas siunčiamas į automobilyje įdiegtą Borto kompiuterį arba šiuo atveju, eksperimentams atlikti, duomenys perduodami į personalinį kompiuterį.

4.6. Tyrimo išvados

1. Atliekant eksperimentinius skaičiavimus nustatyta, jog standartinio koreliacinio metodo realizacija ANSI C kalba neatitinka realaus laiko sistemoms keliamų reikalavimų. Koreliacija su fiksuota aritmetika ir specifinėmis DSP operacijomis paspartino skaičiavimus 460 kartų (iki 2,4 ms) taip priartinant skaičiavimų spartą prie realaus laiko sistemų spartos.
2. Ištirta ar ženklus spartos padidėjimas ir fiksuoto taško aritmetikos naudojimas įneša dideles paklaidas atstumo matavimuose. Maksimalios 1,99% paklaidos santykis su spartos padidėjimu patvirtina, jog galima atsisakyti slankaus taško aritmetikos.

5. Išvados

1. Išanalizavus panašias ir analogiškas sistemas kuriamai sistemai, pasirinkta technologija atstumui matuoti yra ultragarsas, nes jam nėra reikalinga itin sudėtinga ir labai aukštus reikalavimus skaičiavimams tenkinanti techninė įranga (parktronikai). Taip pat įvertinama ir sistemos kaina.
2. Analizuoti pagrindiniai komunikaciniai protokolai naudojami automobiliuose diegiamose kompiuterinėse sistemose, nes atstumo matavimo sistemai tikslinga turėti prieigą prie automobilio stabdžių sistemos.
3. Atlikta atstumo nustatymo metodų apžvalga, kurios metu apspręsta kokių ultragarso bangos signalo apdorojimo principu bus atliekami ultragarso signalo apdorojimas. Pasirinktas *impulso-aido* principas, kuriam reikia paprasčiausios signalo apdorojimo įrangos.
4. Realizuotas atstumo nustatymo algoritmas, kurio pagrindą sudaro koreliacijos metodo taikymas.
5. Atliktas ultragarsu paremto atstumo matavimo matematinio modelio patikrinimas ir pateiktos matavimų ir skaičiavimų fazių algoritmų diagramos.
6. Atliekant matematinio modelio realizaciją DSP platformoje nustatyta, kad naudojamo koreliacijos metodo skaičiavimai trunka ilgai, net 1,06 s. Šios problemos sprendimas – naudoti procesoriaus teikiamas DSP operacijas, kurias taikant koreliacijos algoritmas įvykdomas per 2,4 ms.
7. Atliktas dviejų koreliacijos algoritmo realizacijų palyginimas, kuriame paskaičiavus fiksuoto ir slankaus taško koreliacijos atvejais atstumų santykinę paklaidą gaunama, kad maksimali paklaida 1.99%, kurios santykis su spartos padidėjimu nulemia, kokią koreliacijos algoritmo realizaciją pasirinkti atstumo skaičiavimams.
8. Ultragarso signalo apdorojimą galima atlikti šiuolaikinio mobilaus telefono dydžio įrenginiu.

Literatūra

1. Oica.com. 2011 Production statistics [interaktyvus] [žiūrėta 2014-05-01]. Prieiga per internetą: <http://oica.net/category/production-statistics>
2. Harry Lum and Jerry A. Reagan. Interactive Highway Safety Design Model: Accident Predictive Module [interaktyvus] [žiūrėta 2014-05-01]. Prieiga per internetą: <http://www.fhwa.dot.gov/publications/publicroads/95winter/p95wi14.cfm>
3. The New York Times Company. Google Cars Drive Themselves, in Traffic [interaktyvus]. 2010, [kreiptasi 2014-05-01]. Prieiga per internetą: <http://www.nytimes.com/2010/10/10/science/10google.html?pagewanted=1>
4. AG reporter. Google scores yet another win [interaktyvus]. 2011, [kreiptasi 2014-05-01]. Prieiga per internetą: <http://arabiangazette.com/google-scores-win/>
5. Yahoo! Inc. New Volvo Braking System Hopes to Cut Collisions in Half [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: <http://voices.yahoo.com/article/204418/new-volvo-braking-system-hopes-cut-collisions-in-286493.html?cat=15>
6. Martin Distner, Mattias Bengtsson, Thomas Broberg, Lotta Jakobsson, Volvo Cars, Sweden. CITY SAFETY—A SYSTEM ADDRESSING REAR-END COLLISIONS AT LOW SPEEDS [interaktyvus]. [kreiptasi 2014-05-01]. Prieiga per internetą: <http://www-nrd.nhtsa.dot.gov/pdf/esv/esv21/09-0371.pdf>
7. Volvo Cars. XC60 Features&Equipment [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: <http://www.volvocars.com/intl/all-cars/volvo-xc60/details/pages/features.aspx>
8. Volvo Cars. Build your Volvo [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: <http://www.volvocars.com/us/sales-services/sales/pages/car-configurator.aspx>
9. Bmw.com. Collision Warning with Brake Activation [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: http://www.bmw.com/com/en/insights/technology/connecteddrive/2010/safety/drive_assistance/collision_warning.html
10. The Huffington Post.com, Inc. BMW Self Driving Car: Car maker Shows Off Hands-Free Car On Autobahn [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: http://www.huffingtonpost.com/2012/01/26/bmw-self-driving-car_n_1234362.html
11. Euroncap.com. For safer car crash test safety rating [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: http://www.euroncap.com/rewards/Ford_ActiveCityStop.aspx
12. Media.Ford.com. Adaptive Cruise Control and Collision Warning with Brake Support [interaktyvus]. 2012, [kreiptasi, 2014-05-01]. Prieiga per internetą: http://media.ford.com/images/10031/Adaptive_Cruise.pdf
13. C. Danielson. Mercedes-Benz Tec Day Special Feature: PRE-SAFE And PRE-SAFE Brake [interaktyvus]. 2008 11 12, [kreiptasi 2014-05-01]. Prieiga per internetą: http://www.emercedesbenz.com/Nov08/12_001507_Mercedes_Benz_TecDay_Special_Feature_PRE_SAFE_And_PRE_SAFE_Brake.html
14. Motor Trend. Mercedes-Benz CL-Class Protects Drivers With Pre-Safe System [interaktyvus]. 2006 08 15, [kreiptasi 2014-05-01]. Prieiga per internetą: http://www.motortrend.com/auto_news/112_news060815_mercedes_benz_cl_class_pre_safe_system/view.html
15. Alexander Fuchs. Device and Method for Detecting a Momentary Distance Between a Motor Vehicle and an Obstacle [interaktyvus]. 2007, [kreiptasi 2014-05-01]. Prieiga per internetą: <http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=/netahtml/PTO/srchnum.html&r=1&f=G&l=50&s1=20070273490>
16. Bmw.com. Park distance control [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: http://www.bmw.com/com/en/insights/technology/technology_guide/articles/park_distance_control.html

17. Wikipedia.org. LIDAR [interaktyvus]. 2014, [kreiptasi 2014-05-01]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Lidar>
18. Wikipedia.org. SONAR [interaktyvus]. 2014, [kreiptasi 2014-05-01]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Sonar>
19. Bernd Gombert. Electronic Wedge Brake [interaktyvus]. 2004, [kreiptasi 2014-05-01]. Prieiga per internetą: http://www.dmg-berlin.info/page/downloads/Vortrag_Gombert.pdf
20. Siemens AG. Auto Electronics – Braking Systems [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: http://www.siemens.com/innovation/en/publikationen/publications_pof/pof_fall_2005/auto_electronics/braking_systems.htm
21. How Stuff Works Inc. How brakes work [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: <http://auto.howstuffworks.com/auto-parts/brakes/brake-types/brake1.htm>
22. Wikipedia.org. Engine/Electronic Control Unit [interaktyvus]. 2013, [kreiptasi 2014-05-01]. Prieiga per internetą: http://en.wikipedia.org/wiki/Engine_control_unit
23. Wikipedia.org. CAN bus [interaktyvus]. 2013, [kreiptasi 2014-05-01]. Prieiga per internetą: http://en.wikipedia.org/wiki/CAN_bus
24. Wikipedia.org. LIN [interaktyvus]. 2013, [kreiptasi 2014-05-01]. Prieiga per internetą: http://en.wikipedia.org/wiki/Local_Interconnect_Network
25. Laboratory for Experimental Network and System. ISO9141-2 and J1939 Protocols on OBDII [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: https://www.google.lt/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CEEQFjAA&url=http%3A%2F%2Fmb.xrn.nl%2FDocumentatie%2FDiagnose%2520apparaatuur%2F970911_presentation_02.ppt&ei=yjViU6mMK-qu4QS-x4G4CQ&usq=AFQjCNGz8G14frz6eiyYagRXD4obU8N35w&sig2=fCcbP4Ej_G43M5MXMe4GFQ
26. Wikipedia.org. Keyword Protocol 2000 [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: <http://en.wikipedia.org/wiki/KWP2000>
27. Wikipedia.org. Vehicle bus [interaktyvus]. 2012, [kreiptasi 2014-05-01]. Prieiga per internetą: http://en.wikipedia.org/wiki/Vehicle_bus
28. Peter J. Lagergren, Christian N. Staats, Kenneth M. Wegrzyn, Kelvin L. Woltring, Wayne A. Case. *Ultrasonic liquid level monitoring system*. Proprietor: Schmitt Measurement Systems, Inc. TPK ind.: G01F 23/00. US 8 104 341 B2. 2012-01-31.
29. V. Kazanavičius. Realus laiko neardančiosios kontrolės sistemos sudarymas ir tyrimas. *Daktaro disertacijos santrauka. Technologijos mokslai, informatikos inžinerija (07T)*. 2009, Kaunas
30. Hueber, G., Dept. of Microelectron. Syst., Linz Univ., Austria, Ostermann, T., Bauernfeind, T., Raschhofer, R., Hagelauer, R. New Approach of Ultrasonic Distance Measurement Technique in Robot Applications // Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference. ISBN 0-7803-5747-7, vol. 3, p. 2066-2069, 2000.[INSPEC]
31. W Manthey *et al.* Ultrasonic transducers and transducer arrays for applications in air. *Measurement Science and Technology*. 1995, 3, p. 249-261.
32. Marioli, D. *et al.* Digital Time-of-Flight Measurement for Ultrasonic Sensors // Instrumentation and Measurement Technology Conference, 1991. IMTC-91. 8th Conference Record. May , 14-16, 1991, Atlanta , GA. ISBN 0-87942-579-2. p. 198-201. [INSPEC].
33. T. Ono, M. Kohata, and T. Niyamoto. Ultrasonic phase-sensitive rangefinder with double modulation doppler-free method for shallow seafloor survey. In Proc. IEEE 1984 Ultrasonic Symp., Dallas, TX, NOV. 14-16, 1984
34. P. Kleinschmidt and V. Magori. Ultrasonic remote sensors for non-contact object detection. Siemens Forsch. -u. Entwickl. -Ber., vol. 10, Nr. 2, 1981, pp. 110-118.
35. D. Marioli, E. Sardini, and A. Taroni. Ultrasonic distance measurement for linear and angular position control. IEEE Trans. Instrum. Meas., vol. IM-37, pp. 578-581, Dec. 1988.
36. Guening, F.E. Varlan, M. Eugne, C.E., Dupuis, P. Accurate Distance Measurement by an Autonomous Ultrasonic System Combining Time-of-Flight and Phase-Shift Methods // Instrumentation and Measurement IEEE Transactions. ISSN 0018-9456, vol. 46, no. 6. p. 1236-1240, 1997. [INSPEC].

37. Gregorio Andria, Filippo Attivissimo, Nicola Giaquinto. Digital signal processing techniques for accurate ultrasonic sensor measurement. Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, Bari, Italy. *Elsevier, Measurement*. 2001, 30(2), p. 105-114.
38. Demirli R. and Saniie J.. Model-Based Estimation of Ultrasonic Echoes Part I: Analysis and Algorithms. *IEEE Trans. Ultras., Ferroel. Freq. Control*, vol. 48, n. 3, 2001, p. 787-811.
39. Kazanavičius E. Signalų apdorojimo sistemos. Kaunas. Technologija, 2004.

ANOTACIJA LIETUVIŲ IR ANGLŲ KALBOMIS

Magistrinio darbo tikslas: sukurti automobilio saugaus atstumo palaikymo kompiuterinę sistemą, apimančią atstumą skaičiuojantį algoritmą ir jam realizuoti reikalingą techninės įrangos struktūrą, jos praktinę realizaciją, kuri stebi automobilio atstumą iki priešais esančių objektų (automobilių).

Darbo tikslas įgyvendinamas atliekant egzistuojančių atstumo nustatymo sistemų, komunikacinių protokolų, metodų ultragarso signalui apdoroti, analizę. Sudaromas atstumo nustatymo algoritmas ir matematinis modelis. Atliekamas sudarytojo matematinio modelio diegimas į realią skaičiavimų įrangą, toliau atliekant eksperimentinį tyrimą.

Darbe realizuotas matematinis modelis paremtas koreliacijos metodo taikymu. Algoritmas realizuotas DSP platformoje, kurioje standartinis koreliacijos metodas duoda netinkamas vykdymo trukmes. Realizuota šio metodo modifikacija fiksuota aritmetika DSP operacijomis įvykdoma per tinkamą laiką.

The goal of this thesis is to create automotive safe distance control system which includes distance calculation algorithm and its practical hardware and software realization. System functionality is to measure the distance to the obstacles which are in front of the automobile.

The goal is implemented by analyzing distance measuring systems, communication protocols and methods for ultrasonic signal processing. Furthermore, distance measurement algorithm and its mathematical model was created and installed on its desired hardware for experimental research.

Mathematical model is based on correlation method. Algorithm was run on DSP platform with a standard correlation method which gave unsatisfying execution time results using floating point arithmetic. The modification of the method with fixed point arithmetic and DSP operations was presented and finally gave satisfying execution time results.

Priedai

Priedas A. Atstumo matavimo sistemos matematinis modelis MATLAB

```
function [] = Atstumo_matavimo_matematinis_modelis_MATLAB()
% Nuskaitomi ultragarsinių jutiklių duomenys iš failų:
data100cm = dlmread('100cm_rawData.txt');
datavar50cm = dlmread('variablefrom50cm_rawData.txt');
data110cm = dlmread('110cm_rawData.txt');
Data = [data100cm; datavar50cm; data110cm];
dataSetLen = length(Data(:,1));
timesforStdCor = zeros(1,dataSetLen);
% Nuskaitoma 1-oji failo eilutė (etaloninis signalas ref):
Ints = toUInt16(Data(1, 5:504));
ref = uint16(Ints(31:250));
mref = mean(ref);
% etaloninio signalo atstumas:
etal_atstumas = 1; % atstumas [metrais]
global PLOTLOOP;
PLOTLOOP = 1;
% Garso greitis ore [m/s]:
c = 340.29;
% Diskretus laiko periodas:
dt = 10^-4;
% Laiko ašies paruošimas signalų atvaizdavimui:
t = 0:dt:(220*10^-4);
reflength = length(ref);
revt = fliplr(t(1:reflength));
time_axis = [-revt(1:end-1), t(1:reflength)];
% Pagrindinis matematinio modelio realizacijos ciklas:
for i=1:numel(Data(:,1)),
    % apdorojamas eilinis signalas iš failo:
    Ints = toUInt16(Data(i, 5:504));
    rmeas = uint16(Ints(31:250));
    mrmeas = mean(rmeas);
    subplot(2,1,1),
    plot(t(1:reflength), ref, '-', t(1:reflength), rmeas, ':'),
    xlabel('t, s'),
    ylabel('Signalo energija'),
    title(sprintf('Koreliuojami signalai. refvid= %d , measvid= %d',...
        mref, mrmeas));
    tic;
    % Standartinės koreliacijos realizacija C kalboje:
    ncorr = koreliacijac(ref, rmeas);
    timesforStdCor(i)= toc;
    [~, ind] = max(ncorr);
    %% Koreliacijos rezultato atvaizdavimas ir atstumo paskaičiavimas:
    subplot(2,1,2),
    plot(time_axis, ncorr),
    xlabel('t, s'),
    ylabel('Std. R(d)'),
    title(sprintf('Velinimas: %0.7f s. Atstumas: %0.2f m.',...
        time_axis(ind) , (c*dt*(ind-220)*0.5)+etal_atstumas));
    pause(0.0001);
end
%Išvedamos grafiškai koreliacijos vykdymo trukmės:
figure(2),
plot(timesforStdCor);
% Paskaičiuojamas ir išvedamas 1500-ų koreliacijų
% skaičiavimo trukmių vidurkis:
std_cor_times_mean = mean(timesforStdCor);
```



```
display(std_cor_times_mean);
end
```

Koreliacijac.c:

```
void koreliacijac(uint16_T * x, uint16_T * y , float * res)
{
    int i,j; // Iteracijoms per masyvu reiksmes
    int delay, maxdelay;
    float gmx,gmy,gsx,gsy,gsxy,gdenom;
    maxdelay = calcDataLen -1;
    /* Calculate the mean of the two series x[], y[] */
    gmx = 0.0; //g in variable means global
    gmy = 0.0;
    for (i=0;i<calcDataLen;i++) {
        gmx += x[i];
        gmy += y[i];
    }
    gmx /= calcDataLen;
    gmy /= calcDataLen;
    /* Calculate the denominator */
    gsx = 0.0;
    gsy = 0.0;
    for (i=0;i<calcDataLen;i++) {
        gsx += (x[i] - gmx) * (x[i] - gmx);
        gsy += (y[i] - gmy) * (y[i] - gmy);
    }
    gdenom = (float)sqrt((double)gsx*gsy);
    /* Calculate the correlation series */
    for (delay=-maxdelay;delay<=maxdelay; delay++) {
        gsxy = 0.0;
        i = abs(delay);
        j = 0;
        while((i<=maxdelay)&&(j<=maxdelay)){
            if (delay <= 0){
                gsxy += (x[i] - gmx) * (y[j] - gmy);
            }
            else{
                gsxy += (x[j] - gmx) * (y[i] - gmy);
            }
            i++;
            j++;
        }
        res[delay+calcDataLen-1] = gsxy / gdenom;
    }
}
```

Priedas B. DSPIC33FJ64MC802 mikroprocesoriaus programos kodas

Programą sudarantys moduliai: Main, MyUart, Timer1, CorelationAlg.

Main modulyje konfigūruojama mikroprocesoriaus techninė įranga ir pateikiamas pagrindinis programos ciklas. **MyUart** modulis apibrėžia duomenų siuntimo priėmimo protokolą per RS-232 sąsają ir atlieka pagrindinius atstumo matavimo algoritmo skaičiavimus. **Timer1** apibrėžia mikroprocesoriaus laikmačio (Timer1) panaudojimą atstumo matavimo algoritmo trukmei pamatuoti. **CorelationAlg** pateikia koreliacijos metodo realizaciją slankaus taško aritmetikai.

Main modulis

```
#include <uart.h>
#include <string.h>
#include <stdlib.h>
#include <stdint.h>
#include "myuart.h"
#include "Timer1.h"
#include <p33FJ64MC802.h>
#include "corelationAlg.h"
#include <math.h>

#ifdef __PIC24HJ64GP502__
_FOSCSEL(FNOSC_FRC); // set oscillator mode for FRC ~ 8 Mhz
_FOSC(FCKSM_CSDCMD & OSCIOFNC_ON & POSCMD_NONE); // use OSCIO pin for RA3
_FWDT(FWDTEN_OFF); // turn off watchdog
#elif defined(__dsPIC33FJ64MC802__)
_FOSCSEL(FNOSC_PRIPLL & IESO_OFF); // Internal FRC start-up with PLL,
_FOSC(FCKSM_CSECMD & OSCIOFNC_OFF & POSCMD_XT); // Clock switch disabled,
// Primarily Oscillator XT
_FWDT(FWDTEN_OFF); // Watchdog Timer disabled
_FPOR(FPWRT_PWR128); // Power-up Timer enabled 128 ms
_FICD(JTAGEN_OFF); // Disable JTAG
#endif

void initClockPriWithPLL(unsigned int fbd) {
    // Configure PLL prescaler, PLL postscaler, PLL divisor
    PLLFBD = fbd; // M = fbd+2
    CLKDIVbits.PLLPOST = 0; // N2 = 2
    CLKDIVbits.PLLPRE = 0; // N1 = 2
    // Initiate Clock Switch to Primary Oscillator with PLL (NOSC = 0b011)
    __builtin_write_OSCCONH(0x03);
    __builtin_write_OSCCONL(OSCCON | 0x01);
    // Wait for Clock switch to occur
    while (OSCCONbits.COSC != 0b011);
    // Wait for PLL to lock
    while (OSCCONbits.LOCK != 1) {
    };
}

// Main program with infinite loop :)
int main(void) {
    initClockPriWithPLL(38);
    TRISBbits.TRISB15 = 0; // Set Pin RB15 as output
    LATBbits.LATB15 = 0; // Turn off led on Pin RB15
    init_UART1(86); // Initialize UART1 for 115200,8,N,1 TX/RX and
Interrupts
    CORCON = 0x10F1; // DSP variklio konfiguracija
    start_UART1();
    // delay_ms(500); // delay of 0.5s
    while (FOREVER){ // Do this forever
```

```

        mainWhile();
    }
}

```

Myuart modulis

```

#ifndef MYUART_H
#define MYUART_H

#ifdef __cplusplus
extern "C" {
#endif

#include <p33FJ64MC802.h>
#include "corelationAlg.h"
#include "Timer1.h"
#include <stdio.h>
#include <stdint.h>

#define headerSize sizeof(header)
#define footerSize sizeof(footer)
#define NumOfBuffers 2
#define dataSize 500
typedef union{
    uint8_t rawData[dataSize];
    uint16_t intData[dataSize/2];
}Data_t;
#define NoErrors 0
typedef enum{
    rxHeader,
    rxData,
    rxFooter,
    rxDone
}intt_state_t;

extern unsigned char header[];
extern unsigned char footer[];
extern unsigned char * pBuff;
extern unsigned char actBuf;
extern intt_state_t rx_State;

extern void init_UART1( int BRGVAL );
extern void processData( Data_t * DataBlock );
extern void mainWhile( void );
extern void start_UART1( void );
extern void stop_UART1( void );
extern void sendString(char *sptr);

#ifdef __cplusplus
}
#endif
#endif /* UART_H */

```

Myuart.c:

```

#include "myuart.h"

unsigned char header[] = {0xFF, 0x01, 0x00, 0x00};
unsigned char footer[] = {0xFF, 0xFF, 0x00, 0x01};

unsigned char actBuf = 0;

```

```

intt_state_t rx_State = rxHeader;

Data_t Buff1 __attribute__((space(xmemory))); //foreference signal
Data_t Buff2 __attribute__((space(ymemory)));
Data_t Buff3 __attribute__((space(ymemory)));
unsigned char * pBuff = Buff1.rawData;
Data_t * dataPack;

void __attribute__((interrupt, no_auto_psv)) _U1RXInterrupt(void) {
    static unsigned int poz;
    unsigned char newByte = U1RXREG; // nuskaitom baita
    // U1STAbits.OERR = 0;
    switch (rx_State) {
        case rxHeader:
            if (newByte == header[poz]) {
                if (++poz == headerSize) {
                    rx_State = rxData;
                    poz = 0;
                }
            } else poz = 0;
            break;
        case rxData:
            pBuff[poz] = newByte;
            if (++poz == dataSize) {
                poz = 0;
                rx_State = rxFooter;
            }
            break;
        case rxFooter:
            if (newByte == footer[poz]) {
                if (++poz == footerSize) {
                    rx_State = rxDone;
                    poz = 0;
                }
            } else poz = 0;
            break;
        case rxDone:
            break;
        default:
            poz = 0;
    }
    IFS0bits.U1RXIF = 0;
}

void __attribute__((interrupt, no_auto_psv)) _U1TXInterrupt(void) {
    IFS0bits.U1TXIF = 0;
}

// Funkcija inicializuojanti UART1 moduli
void init_UART1(int BRGVAL)
{
    OSCCON = 0x46;
    OSCCON = 0x57;
    OSCCONbits.IOLOCK = 0;

    // RP register assignment:
    RPOR5bits.RP11R = 3; // Assign U1TX peripheral to pin RP11
    RPINR18bits.U1RXR = 10; // Assign pin RP10 to U1RX peripheral

    // UART Registers
    // dspic33f reference manual 17 page 7

```

```

    // Configure UART Mode Register
    U1MODEbits.UARTEN = 0; // Bit15 - Initially disable, enable at end of
function.
    U1MODEbits.USIDL = 0; // Bit13 - Continue in Idle
    U1MODEbits.IREN = 0; // Bit12 - No IR Translation
    U1MODEbits.RTSMD = 0; // Bit11 - Simplex Mode
    U1MODEbits.UEN = 0; // Bits8,9 - TX,RX enabled, CTS,RTS not
    U1MODEbits.WAKE = 0; // Bit7 - No Wake Up (since we don't sleep here)
    U1MODEbits.LPBACK = 0; // Bit6 - No Loop Back
    U1MODEbits.ABAUD = 0; // Bit5 - No Autobaud
    U1MODEbits.URXINV = 0; // Bit4 - IdleState = 1 (for dsPIC)
    U1MODEbits.BRGH = 1; // Bit3 - 16 clocks per bit period
    U1MODEbits.PDSEL = 0; // Bits1,2 - 8bit, No Parity
    U1MODEbits.STSEL = 0; // Bit0 One Stop Bit

    U1BRG = BRGVAL; // Set Baudrate based on value of BRGVAL

    // Configure UART Status and Control Register
    U1STAbits.UTXISEL1 = 0; // Bit15 - Interrupt when Char is transfered
(1/2 config!)
    U1STAbits.UTXINV = 0; // Bit14 - N/A, IRDA config
    U1STAbits.UTXISEL0 = 0; // Bit13 - Other half of Bit15
    U1STAbits.UTXBRK = 0; // Bit11 - Disabled
    U1STAbits.UTXEN = 1; // Bit10 - TX disabled
    IEC0bits.U1RXIE = 1; // Enable RX interrupts
    IEC0bits.U1TXIE = 1; // Disable TX interrupts
    U1STAbits.URXISEL = 0; // Bits6,7 - Interrupt when a character is
recieved
    U1STAbits.ADDEN = 0; // Bit5 - Address Detect Disabled
}

void sendData(char *cptr, int n) // Function to send char data
{
    while (n) {
        while (U1STAbits.UTXBF == 1); // Wait until TX buf read for new data
        U1TXREG = *cptr;
        cptr++;
        n--;
    }
}

unsigned int GoodPacks = 0;
volatile unsigned int * ref; // ref - signalas kurio at?vilgiu bus
skai?iuojama koreliacija
volatile unsigned int * meas; // eilinis pamatuotas signalas, kuris bus
koreliuojamas su ref;
unsigned int dataToIgnore = 30; //Ignoruojama 30 16-bitu reiksmiu
int dataToSendLen;
//volatile unsigned int atstumai[1500];
//unsigned int zz = 0;
#ifdef MEASURE_EXECUTION_TIME
volatile long double timStdCor = 0.0L;
#endif

void processData(Data_t * DataBlock) {

    GoodPacks++;
    if (GoodPacks==1){
        // Paimti reference signalo paketa ref:
        ref = DataBlock->intData;
        ref = ref + dataToIgnore;
        return;
    }
}

```

```

    }else{
        // Eilinio matavimo duomenys:
        meas = DataBlock->intData;
        meas = meas + dataToIgnore;

#ifdef MEASURE_EXECUTION_TIME
        // Koreliavimo trukmes paskaiciavimas:
        Timer1InitForDurationCalc();
        T1CONbits.TON = 1; // Paleidzia Timer1 ir skaiciuoja
#endif

        // Testuojamas kodo gabalas

#ifdef REZ_FLOAT
        std_cross_corelation(ref, meas);
#elif defined(REZ_UINT_32)
        std_cross_corelation_DSP(ref, meas);
#endif

#ifdef MEASURE_EXECUTION_TIME
        T1CONbits.TON = 0; // Sustabdo Timer1
        timStdCor = (long double)((PERIOD * Counter) + TMR1)*TCY;
#endif

#if defined( SEND_CORELATION_RESULT_TO_PC) &&(!defined(SHORT_COR_REZ))
#ifdef REZ_FLOAT
        dataToSendLen = (int)resultDataLen * sizeof(float);
#elif defined(REZ_UINT_32)
        dataToSendLen = (int)resultDataLen * sizeof(uint32_t);
#endif

#ifdef REZ_FLOAT
        sendData(header, (int)sizeof(header));
        sendData((char*)&R, dataToSendLen);
        sendData(footer, (int)sizeof(footer));
#elif defined(REZ_UINT_32) && (!defined(SHORT_COR_REZ))
        sendData(header, (int)sizeof(header));
        sendData((char*)&suma, dataToSendLen);
        sendData(footer, (int)sizeof(footer));
#endif

#endif

#ifdef MEASURE_EXECUTION_TIME) && defined(SEND_TIMES)
        sendData((char*)&timStdCor, (int)sizeof(long double));
#endif

#if defined(SEND_DISTANCES) && defined(REZ_FLOAT)
        sendData((char*)&D, (int)sizeof(float));
#elif defined(SEND_DISTANCES) && defined(REZ_UINT_32)
        sendData((char*)&DD, (int)sizeof(uint32_t));
#endif
    }
}

void mainWhile(void) {
    if (rx_State == rxDone) {
        LATBbits.LATB15 = 1;
        dataPack = (Data_t *) pBuff; // pirmas buferis Buff1
        if (pBuff == Buff1.rawData){
            pBuff = Buff2.rawData;

```

```

        }else{
            if(pBuff == Buff2.rawData){
                pBuff = Buff3.rawData;
            }else{
                pBuff = Buff2.rawData;
            }
        }
        rx_State = rxHeader;
        proccessData(dataPack);
        LATBbits.LATB15 = 0;
    }
}

void start_UART1(void) {
    U1MODEbits.UARTEN = 1;
    U1STAbits.UTXEN = 1;
}

void stop_UART1(void) {
    U1MODEbits.UARTEN = 0;
    U1STAbits.UTXEN = 0;
}

void sendString(char *sptr)
{
    while (*sptr != 0) {
        while (U1STAbits.UTXBF == 1);
        U1TXREG = *sptr;
        sptr++;
    }
}

```

Timer1 modulis

```

#ifndef TIMER1_H
#define TIMER1_H

#ifdef __cplusplus
extern "C" {
#endif
#define TCY 0.000000025L // 40 MHz instrukciju vykdymo daznio periodas
#define PERIOD 0xFFFF // Laikmacio skaitliuko periodas
#define FOREVER 1
#include <stdint.h>
#include <p33FJ64MC802.h>
extern volatile uint32_t Counter;
extern void Timer1InitForDurationCalc( void );
#ifdef __cplusplus
}
#endif

#endif /* TIMER1_H */

```

Timer1.c:

```

#include "Timer1.h"
#include <stdint.h>

volatile uint32_t Counter;

void __attribute__((interrupt, no_auto_psv)) _T1Interrupt(void) {
    Counter++;
}

```

```

    IFS0bits.T1IF = 0;
}

void Timer1InitForDurationCalc(void) {
    T1CON = 0;
    Counter = 0;
    TMR1 = 0;
    IFS0bits.T1IF = 0; /* reset Timer 1 interrupt flag */
    IPC0bits.T1IP = 1; /* set Timer1 interrupt priority level to 4 */
    IEC0bits.T1IE = 1; /* enable Timer 1 interrupt */
    PR1 = PERIOD; /* set Timer 1 period register */
    T1CONbits.TCKPS = 0; /* select Timer1 Input Clock Prescale */
    T1CONbits.TCS = 0; /* select Internal clock (FOSC/2) */
}

```

Koreliacijos algoritmo modulis

```

#ifndef CORELATIONALG_H
#define CORELATIONALG_H

#ifdef __cplusplus
extern "C" {
#endif
#include <math.h>          // Standartines matematinės funkcijos
#include "Timer1.h"       // Matuosim koreliacijos algoritmo vykdymo trukmes
#include <stdint.h>
#include <dsp.h>

#define calcDataLen (220)
#define resultDataLen ((2*calcDataLen)-1)

// #define REZ_FLOAT
#define REZ_UINT_32
#define SHORT_COR_REZ /* Koreliacijos rezultato reikšmes nededamos i
masyva*/

// #define MEASURE_EXECUTION_TIME
// #define SEND_TIMES
// #define SEND_CORELATION_RESULT_TO_PC
// #define SEND_DISTANCES

#define SSIA (340.29) /* Sound Speed In Air */

#ifdef REZ_FLOAT
#define DT (0.0001)
#define REF_DISTANCE (1) /* reference distance in meters*/
#elif defined(REZ_UINT_32)
#define DT (0.01)
#define REF_DISTANCE (100) /* reference distance in centimeters*/
#endif

#define POINT_FIVE (0.5)
#define DISTANCE_COEF (SSIA*DT*POINT_FIVE)

#ifdef REZ_FLOAT
extern volatile float R[];
extern volatile float D;

#elif defined(REZ_UINT_32)
#ifdef SHORT_COR_REZ
extern uint32_t suma;
#else
extern uint32_t suma[];

```



```

#endif
extern uint32_t DD;
#endif
extern void std_cross_corelation(volatile unsigned int * x,
                                volatile unsigned int * y);
extern void MeanVar(volatile unsigned int *ptrArr, unsigned int Len,
                  unsigned int *ptrMean);
extern void multAcc(volatile unsigned int * mas1, volatile unsigned int *
mas2,
                  volatile unsigned int len);
extern void std_cross_corelation_DSP(volatile unsigned int * x,
                                    volatile unsigned int * y);

#ifdef __cplusplus
}
#endif

#endif /* CORELATIONALG_H */

```

CorelationAlg.c:

```

#include "corelationAlg.h"
#ifdef REZ_FLOAT
volatile float R[resultDataLen];
volatile float maksVal;
volatile float gmx, gmy, gsx, gsy, gsxy, gdenom;
volatile float D;
#elif defined(REZ_UINT_32)
#ifdef SHORT_COR_REZ
uint32_t suma;
uint32_t maxVal;
uint32_t DD;
#else
uint32_t suma[resultDataLen];
uint32_t maxVal;
uint32_t DD;
#endif
#endif
// Bendri koreliacijos skaiciavimo parametrai
#define maxdelay (calcDataLen-1)
int i, j;
volatile int delay;
volatile int maxIndDelay;
unsigned char first_max = 1;

#ifdef REZ_FLOAT
void std_cross_corelation( volatile unsigned int * x,
                          volatile unsigned int * y)
{
    //Skaiciuojami koreliuojamu seku vidurkiai:
    gmx = 0.0;
    gmy = 0.0;
    for (i=0; i<calcDataLen; i++) {
        gmx += x[i];
        gmy += y[i];
    }
    gmx /= calcDataLen;
    gmy /= calcDataLen;
    // Skaiciuojama koreliacijos seku dispersija
    gsx = 0.0;
    gsy = 0.0;
    for (i=0; i<calcDataLen; i++) {

```

```

        gsx += (x[i] - gmx) * (x[i] - gmx);
        gsy += (y[i] - gmy) * (y[i] - gmy);
    }
    gdenom = sqrt(gsx*gsy);
    /* Koreliacines sumos kaupimas */
    for (delay=-maxdelay;delay<=maxdelay;delay++) {
        gsxy = 0.0;
        i = abs(delay);
        j = 0;
        while((i<=maxdelay)&&(j<=maxdelay)){
            if (delay <= 0){
                gsxy += (x[i] - gmx) * (y[j] - gmy);
            }
            else{
                gsxy += (x[j] - gmx) * (y[i] - gmy);
            }
            i++;
            j++;
        }
        R[delay+calcDataLen-1] = gsxy / gdenom;
        // Maksimumo fiksavimas:
        if(first_max){
            maksVal = R[delay+calcDataLen-1];
            maxIndDelay = delay;
            first_max = 0;
        } else{
            if(R[delay+calcDataLen-1] >= maksVal){
                maksVal = R[delay+calcDataLen-1];
                maxIndDelay = delay;
            }
        }
    }
    first_max = 1;
    // Atstumo paskaiciavimas:
    D = DISTANCE_COEF*maxIndDelay + REF_DISTANCE;
}
#endif
// Paskaiciuojamas masyvo ptrArr, kurio ilgis Len, vidurkis ptrMean
/*void MeanVar(volatile unsigned int *ptrArr, unsigned int Len,
               unsigned int *ptrMean) {

    uint32_t suma = 0;
    asm("mov [W14+4], W10");           // W10 = ptrArr
    asm("mov [W14+6], W7");           // W7 = Len
    asm("sub W7, #1, W2");            // W2 = Len-1

    asm("clr A");
    asm("repeat W2");
    asm("add [W10++], #0, A");         // A = sum(arr)

    asm("add W7, #1, A");
    // A = A + (Len/2) for div's lack of rounding ...
    suma |= ACCAU;
    suma = suma << 16;
    suma |= ACCAH;
    *ptrMean = suma/calcDataLen;
}*/
/*****
***** Koreliacija su fiksuoto tasko aritmetika *****/
void multAcc(volatile unsigned int * mas1, volatile unsigned int * mas2,
             volatile unsigned int len)
{

```

```

    asm("mov [W14], W8");
    asm("mov [W14+2], W10");
    asm("mov [W14+4], W2");
    asm("sub W2, #1, W2");
    asm("mov #0, W4");
    asm("mov #0, W6");
    asm("repeat W2");
    asm("MAC W4*W6, A, [W8]+=2, W4, [W10]+=2, W6");
    asm("MAC W4*W6, A");
}

#ifdef REZ_UINT_32

#ifdef SHORT_COR_REZ
void std_cross_corelation_DSP( volatile unsigned int * x,
                               volatile unsigned int * y)
{
    maxVal = 0;
    for (delay=-maxdelay;delay<=maxdelay;delay++) {
        suma = 0;
        i = abs(delay);
        j = 0;
        // Koreliacines sumos kaupimas:
        if (delay <= 0){
            multAcc(&x[i], &y[j], maxdelay-i+1);
        }
        else{
            multAcc(&x[j], &y[i], maxdelay-i+1);
        }
        suma |= ACCAH;
        suma = suma << 16;
        suma += ACCAL;
        asm("clr A");
        // Maksimumo fiksavimas:
        if(suma >= maxVal){
            maxVal = suma;
            maxIndDelay = delay;
        }
    }
    // Atstumo paskaiciavimas:
    DD = DISTANCE_COEF*maxIndDelay + REF_DISTANCE;
}

#else

void std_cross_corelation_DSP( volatile unsigned int * x,
                               volatile unsigned int * y)
{
    maxVal = 0;
    for (delay=-maxdelay;delay<=maxdelay;delay++) {
        suma[delay+calcDataLen-1] = 0;
        i = abs(delay);
        j = 0;
        // Sumos kaupimas:
        if (delay <= 0){
            multAcc(&x[i], &y[j], maxdelay-i+1);
        }
        else{
            multAcc(&x[j], &y[i], maxdelay-i+1);
        }
        suma[delay+calcDataLen-1] |= ACCAH;
        suma[delay+calcDataLen-1] = suma[delay+calcDataLen-1] << 16;
        suma[delay+calcDataLen-1] += ACCAL;
    }
}

```

```

asm("clr A");
// Maksimumo fiksavimas:
if(suma[delay+calcDataLen-1] >= maxVal){
    maxVal = suma[delay+calcDataLen-1];
    maxIndDelay = delay;
}
}
// Atstumo paskaiciavimas:
DD = DISTANCE_COEF*maxIndDelay + REF_DISTANCE;
}
#endif
#endif
/*****

```

Priedas C. Eksperimentų vykdymo schemos realizacija MATLAB

```

%% Programa, kuri atlieka:
% -duomenų nuskaitymą iš failų po eilutę;
% -siunčia tuos duomenis kaip paketą per COM portą į dsPIC33F procesorių
%%
function [] = writeDataFromFileTodsPIC33FdiagInfolast()
% Sukonfigūruoti RS232 sąsają 508 baitų siuntimui
ports = instrfind;
i = length(ports);
if(i ~= 0)
    for j = 0:i-1
        if strcmp(ports(i-j).Status, 'open'),
            fclose(ports(i-j));
        end
        delete(ports(i-j));
    end
end
port = 'COM1';
obj=serial(port);

obj.BaudRate=115200;

obj.ByteOrder='littleEndian';
% %%%%%%%%%%% Programos switchai %%%%%%%%%%%
isvestiGrafika = 0;
% %%%%%%%%%%%
% % float/uint32 baitu kiekis:
obj.InputBufferSize=1764; % in bytes
obj.BytesAvailableFcnCount=1764;
%% double baitu kiekis:
obj.InputBufferSize=8; % in bytes
obj.BytesAvailableFcnCount=8;
% obj.InputBufferSize=25; % in bytes
% obj.BytesAvailableFcnCount=25;
obj.OutputBufferSize = 508; % 20140204
obj.BytesAvailableFcnMode='byte';

obj.BytesAvailableFcn = {@getNewData};
obj.ReadAsyncMode = 'continuous';
obj.Timeout = 5;
% obj.ErrorFcn = {@time};

if strcmp(obj.Status, 'closed'),
    fopen(obj);

```

```

end

global PLOTLOOP; PLOTLOOP=1;
global it; it = 1;
global picData; picData = [];
% Nuskaitome visus duomenis esančius failuose
data100cm = dlmread('100cm_rawData.txt');
datavar50cm = dlmread('variablefrom50cm_rawData.txt');
data110cm = dlmread('110cm_rawData.txt');
Data = [data100cm; datavar50cm; data110cm];
% pasiruosiam reference signala koreliacijai:
Ints = toUInt16(Data(1, 5:504));
ref = uint16(Ints(31:250));
% Pasiruošimas cikliškai koreliuoti:
rowCount = length(Data(:,1));
El_num = rowCount;
dt = 0.0001;
delayTime = (-219:1:219)*dt; % GRAfikui laiko ašis
time = delayTime(220:end);
obj.UserData.isNew=0;
global Info;
Info = zeros(1, rowCount);
while rowCount,
    R = [];
    %% siunciam duomenis i PIC'a:
    cycle_Cnt = El_num-rowCount+1;
    fwrite(obj, Data(cycle_Cnt,:), 'uint8');
    %% konvertuojam paketo duomenis i 16 bitu duomenis:
    Ints = toUInt16(Data(El_num-rowCount+1, 5:504));
    IntsForCor = Ints(31:250);

    %% Paverčiame matlab double reikšmių masyvą į uint16:
    IntsForCor16bit = uint16(IntsForCor);
    %% koreliacijos patikrinimui R lyginamas su paskaiciuotu PIC'e:
    if(El_num-rowCount ~= 0),
        R = koreliacijac(ref, IntsForCor16bit); % Realizacija A priede
    end
    %% Palyginimui išvedami grafikai:
    if isvestiGrafika,
        if(~isempty(R) && ~isempty(picData)),
            %
            subplot(3,1,1),
            plot(time, ref, '-', time, IntsForCor16bit, ':'),
            xlabel('t, s'),
            ylabel('Signalų energija'),
            title(sprintf('Koreliuojami signalai.'));

            subplot(3,1,2),
            plot(delayTime, R),
            xlabel('Vėlinimai, s'),
            ylabel('Std. R(d)'),
            title(sprintf('Koreliacijos rezultatas MATLAB su FLOAT tipu.'));

            subplot(3,1,3),
            plot(delayTime, picData(2:end-1)),
            xlabel('Vėlinimai, s'),
            ylabel('Std. R(d)'),
            title(sprintf('Koreliacijos rezultatas iš dsPIC33F su UINT32
tipu.'));

            end
        end
    end
end

```

```
        rowCount = rowCount - 1;
end
disp('Siuntimas baigtas... ');
fclose(obj);
disp(obj);
delete(obj);
clear obj;
end

function getNewData(obj,~)
global Info;
global it;
[Info(it), ~] = fread(obj, 1, 'double');
it = it + 1;
end
```