



Kauno technologijos universitetas

Informatikos fakultetas

**Automatizuoto sandėlio roboto su vedliu judėjimo tikslumo
priklausomybės nuo skirtingų komponentų ir metodų tyrimas**

Baigiamasis magistro projektas

Mantas Dulinskas

Projekto autorius

Lekt. Dominykas Barisas

Vadovas

Kaunas, 2019



Kauno technologijos universitetas

Informatikos fakultetas

**Automatizuoto sandėlio roboto su vedliu judėjimo tikslumo
priklausomybės nuo skirtingų komponentų ir metodų tyrimas**

Baigiamasis magistro projektas

Programų sistemų inžinerija (6211BX011)

Mantas Dulinskas

Projekto autorius

lekt. Dominykas Barisas

Vadovas

m. d. dr. Mantas Lukoševičius

Recenzentas

Kaunas, 2019



Kauno technologijos universitetas

Informatikos fakultetas

Mantas Dulinskas

Automatizuoto sandėlio roboto su vedliu judėjimo tikslumo priklausomybės nuo skirtingų komponentų ir metodų tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Manto Dulinsko, baigiamasis projektas tema „Automatizuoto sandėlio roboto su vedliu judėjimo tikslumo priklausomybės nuo skirtingų komponentų ir metodų tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

(vardą ir pavardę įrašyti ranka)

(parašas)



Kauno technologijos universitetas

Informatikos fakultetas

Baigiamojo magistro projekto užduotis

Projekto tema	Automatizuoto sandėlio roboto su vedliu projektavimas ir valdymo sistemos kūrimas
Reikalavimai ir sąlygos (tikslinti pavadinimą pagal poreikį)	Užduotis ir tyrimo tikslas – suprojektuoti roboto prototipą, kuris sugebėtų automatiškai judėti uždaroje patalpoje sekdamas liniją ir orientuotis erdvėje pagal ant žemės sudėliotas RFID žymes bei parašyti roboto valdymo programinę įrangą skirtą roboto darbo valdymui bei komunikacijai su kitais įrenginiais.
Vadovas / Vadovė	Lekt. Dominykas Barisas (vadovo pareigos, vardas, pavardė, parašas) (data)

Dulinskas, Mantas. Automatizuoto sandėlio roboto su vedliu judėjimo tikslumo priklausomybės nuo skirtingų komponentų ir metodų tyrimas. Magistro baigiamasis projektas / vadovas lekt. Dominykas Barisas; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų kryptių grupė): Programų sistemų inžinerija.

Reikšminiai žodžiai: Automatinis robotas, automatizuotas sandėlio robotas, robotika, logistika, linijos sekimas, AGV, AMV, RFID, planavimo sistema

Kaunas, 2019. 70 p.

Santrauka

Šiame magistro darbe buvo projektuojamas automatizuotam sandėliui skirtas roboto prototipas ir kuriama jo užduočių bei komponentų valdymo programinė įranga. Siekiamybė buvo sukurti robotą, kuris gebėtų be žmogaus įsikišimo judėti uždaroje sandėlio patalpoje sekant ant žemės nubrėžtą liniją ir pasiekti reikiamą užduoties tašką orientuojantis pagal linijoje esančias žymes. Taip pat programinė įranga turėjo gebėti valdyti išorinį prekių paėmimo įrenginį, skenuoti lentynas bei realiu laiku reaguoti į jutiklių duomenis.

Darbo metu buvo suprojektuotas ir iš mėgėjiškų komponentų surinktas roboto prototipas. Jis buvo naudojamas misijų programinės įrangos testavimui realiu laiku. Roboto mechaninės dalies valdymas buvo realizuotas panaudojant Arduino mikrokontrolerį programuojant C++ kalba, o pagrindinė misijų valdymo sistema suprogramuota Node.js vykdymo aplinkoje ir įdiegta RaspberryPi kompiuteryje.

Kadangi vienas svarbiausių faktorių tolimesniam projekto vystymui yra judėjimo tikslumas, tiriamojo darbo metu buvo eksperimentuojama su skirtingais roboto nustatymais, kad išsiaiškinti technines ir programines galimybes tikslumo pagerinimui. Taip pat buvo optimizuojamas automatinio skenavimo algoritmas skirtas iš gautų duomenų apskaičiuoti sandėlio lentynų dydžius ir pozicijas.

Dulinskas, Mantas. Research of Warehouse Automated Guided Vehicle Movement Accuracy Depending on Different Components and Methods. Master's Final Degree Project / supervisor lect. Dominykas Barisas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Software engineering.

Keywords: Autonomous robot, automated guidance vehicle, robotics, warehouse, logistic, line tracing, AGV, AMV, RFID, mission planner.

Kaunas, 2019. 70 pages.

Summary

In this master thesis I designed robot prototype for automated warehouse and developed it's task management software. The aim was to create a robot that would be able to move in a closed warehouse without human intervention, following the line drawn on the ground and reaching the required point of the target, guided by the line-based markers. Also, the software had to be able to manage the external pickup device, scan the shelves, and react to the sensor data in real time.

The robot's prototype from the amateur components was designed during the work. It was used for real-time mission software testing. The control of the robot's mechanical part was realized by using Arduino microcontroller in C ++ programming, while the main mission management system was programmed in the Node.js execution environment and installed on the RaspberryPi computer.

As one of the most important factors for the further development of the project is the accuracy of the movement, during the experimental work I experimented with different robot settings to clarify the technical and software capabilities to improve accuracy. The automatic scanning algorithm was also optimized for calculating warehouse shelf sizes and positions from the data received.

Turinys

Lentelių sąrašas	9
Paveikslų sąrašas	10
Santrumpų ir terminų sąrašas	12
1. Įvadas.....	13
1.1. Dokumento paskirtis.....	13
1.2. Projekto tikslas	13
1.3. Uždaviniai.....	14
1.4. Užsakovas.....	14
2. Analitinė dalis	15
2.1. Analizės tikslas.....	15
2.2. Robotų tipai	15
2.2.1. Automatizuoto vedžiojimo transporto priemonės	15
2.2.2. Autonominis mobilusis robotas	17
2.2.3. Robotų tipų palyginimas.....	18
2.3. AGV judėjimo metodai	19
2.3.1. Magnetinė juosta	19
2.3.2. Šviesą atspindinčios juostos	21
2.3.3. Lazerinė trianguliacija	21
2.3.4. Švyturiai	22
2.4. Autonominio judėjimo metodai.....	22
2.4.1. SLAM algoritmai.....	22
2.4.2. Aplinkos skenavimas.....	23
2.5. Metodų palyginimas	23
2.6. Analizės išvados	23
3. Projektinė dalis	24
3.1. Sistemos paskirtis	24
3.2. Projekto kūrimo pagrindimas	24
3.3. Pasirinktas sprendimas	24
3.4. Sistemos panaudos atvejai	25
3.5. Funkciniai reikalavimai	29
3.6. Nefunkciniai reikalavimai	31
3.6.1. Reikalavimai panaudojamumui	31
3.6.2. Reikalavimai vykdymo charakteristikos	31
3.6.3. Reikalavimai veikimo sąlygoms.....	31
3.6.4. Reikalavimai sistemos priežiūrai.....	32
3.7. Sistemos kūrimo rizikos	32
3.8. Sistemos architektūra.....	32
3.8.1. Architektūros pateikimas.....	32
3.8.2. Architektūros tikslai ir apribojimai	32
3.8.3. Sistemos statinis vaizdas	33
3.8.4. Sistemos dinaminis vaizdas	35
3.9. Sistemos testavimas.....	42
3.9.1. Testavimo įrankiai ir aplinka.....	42
3.9.2. Testavimo procedūra	42

3.9.3. Testavimo rezultatai ir išvados	44
3.10. Sistemos kūrimo rezultatas	45
4. Tyrimo dalis	48
4.1. Įvadas	48
4.2. Pasirinkti tyrimai	48
4.3. Tyrimui naudojama įranga	48
4.4. Tyrimas Nr.1.....	49
4.4.1. Aritmetinio vidurkio metodas.....	51
4.4.2. Didžiausios-mažiausios reikšmės metodas.....	52
4.4.3. Dažniausio pasikartojimo metodas.....	52
4.4.4. Apibendrinimas	53
4.5. Tyrimas Nr.2.....	53
4.5.1. Judėjimas be stabdymo.....	53
4.5.2. Palaiptinis greičio mažinimas	54
4.5.3. Lėtėjimas ir variklio blokavimas	55
4.6. Tyrimas Nr.3.....	56
4.6.1. Išsiskleidžiantis paėmimas	56
4.6.2. Magnetinis paėmimas	57
4.6.3. Vakuuminis paėmimas	59
5. Eksperimentinė dalis	60
5.1. Eksperimentas Nr.1	60
5.2. Eksperimentas Nr.2	62
5.3. Eksperimentas Nr.3	64
5.4. Eksperimentų išvados	66
Išvados	67
Literatūros sąrašas	68
Priedai.....	70
1 priedas. Įdiegimo aktas	70

Lentelių sąrašas

1 lentelė. Robotų tipų palyginimas.....	18
2 lentelė. Metodų paklaidų palyginimas	23
3 lentelė. Panaudos atvejo „Pradėti naują maršrutą aprašas“.....	26
4 lentelė. Panaudos atvejo „Stabdyti maršrutą“ aprašas	26
5 lentelė. Panaudos atvejo „Gauti ataskaitą“ aprašas.....	27
6 lentelė. Panaudos atvejo „Peržiūrėti statusą“ aprašas	27
7 lentelė. Panaudos atvejo „Keisti vartotojo parametrus“ aprašas	27
8 lentelė. Panaudos atvejo „Įkelti naują žemėlapi“ aprašas	28
9 lentelė. Panaudos atvejo „Keisti sistemos ir roboto nustatymus“ aprašas	28
10 lentelė. Panaudos atvejo „Surasti poziciją“ aprašas	28
11 lentelė. Funkcinis reikalavimas FR-1.....	29
12 lentelė. Funkcinis reikalavimas FR-2.....	29
13 lentelė. Funkcinis reikalavimas FR-3.....	29
14 lentelė. Funkcinis reikalavimas FR-4.....	30
15 lentelė. Funkcinis reikalavimas FR-5.....	30
16 lentelė. Funkcinis reikalavimas FR-6.....	30
17 lentelė. Reikalavimas panaudojamumui NR-1.....	31
18 lentelė. Reikalavimas vykdymo charakteristikoms NR-2.....	31
19 lentelė. Reikalavimas sistemos priežiūrai NR-3	32
20 lentelė. Sistemos kūrimo rizikos	32
21 lentelė. Priėmimo testavimo rezultatas	44
22 lentelė. Pasirinkti tyrimai	48
23 lentelė. Eksperimento Nr.1 apibendrinti rezultatai	60
24 lentelė. Eksperimento Nr.2 bandymų rezultatai	63
25 lentelė. Eksperimento Nr.3 įrenginių vertinimo kriterijų lentelė	64
26 lentelė. Išsiskleidžiančio paėmimo vertinimo rezultatas.....	65
27 lentelė. Magnetinio paėmimo vertinimo rezultatas	65
28 lentelė. Vakuuminio paėmimo vertinimo rezultatas	66

Paveikslų sąrašas

2.1 pav. Robotams nutiesto kelio pavyzdys [7]	16
2.2 pav. Fiksuotas vienos krypties roboto kelias.....	17
2.3 pav. AMR tipo robotas sandėlio aplinkoje [10].....	17
2.4 pav. SLAM metodu sudarytas aplinkos žemėlapis [11].....	18
2.5 pav. Magnetinio tipo sekimas AGV įrenginiui [12]	20
2.6 pav. Nukrypimo nuo linijos matavimas [12]	20
2.7 pav. Įrenginio pozicijos nustatymas erdvėje pagal lazerio atspindžius [16].....	21
2.8 pav. Absoliutinių ir vietinių koordinatų sistema vietos nustatymui [16]	22
3.1 pav. Tinklinio tipo žemėlapio pavyzdys su numeriais pažymėtomis RFID žymėmis	25
3.2 pav. Roboto valdymo sistemos panaudos atvejų diagrama.....	25
3.3 pav. Abstraktus sistemos skaidymas į paketus.....	33
3.4 pav. „WEB API“ paketas	33
3.5 pav. „Extensions“ paketas.....	34
3.6 pav. „Mission planner“ paketas	34
3.7 pav. „Serial“ paketas	34
3.8 pav. „NFC“ paketas.....	35
3.9 pav. „Settings“ paketas.....	35
3.10 pav. „Laser“ paketas.....	35
3.11 pav. Roboto valdymo sistemos būsenų diagrama	36
3.12 pav. Panaudos atvejo „Pradėti naują maršrutą“ veiklos diagrama.....	37
3.13 pav. Panaudos atvejo „Įkelti naują žemėlapi“ veiksmų diagrama	38
3.14 pav. Lazerio duomenų apdorojimo proceso veiklos diagrama.....	39
3.15 pav. Panaudos atvejo „Pradėti naują maršrutą“ sekų diagrama	40
3.16 pav. Panaudos atvejo „Įkelti naują žemėlapi“ sekų diagrama.....	40
3.17 pav. Roboto valdymo sistemos išdėstymo diagrama	41
3.18 pav. Gretimumo matrica	42
3.19 pav. Dvimatė krypčių matrica	42
3.20 pav. Integracinio testavimo iš apačios į viršų schema	43
3.21 pav. Vienetų testavimo rezultatai	44
3.22 pav. Pirmoji roboto versija skirta važiavimo programinės įrangos testavimui.....	46
3.23 pav. Antrosios versijos roboto prototipas	47
3.24 pav. Antrosios versijos roboto prototipas su uždėtu paėmimo įrenginiu	47
4.1 pav. Skenavimui ir testavimui naudotos lentynos.....	49
4.2 pav. Vertikalios skenavimo schema	50
4.3 pav. Horizontalios skenavimo schema	50
4.4 pav. Celės kraštų skenavimo problema.....	50
4.5 pav. Skenavimo metu gautų koordinatų pavyzdys.....	51
4.6 pav. Metodo norimų gauti reikšmių vizualizacija.....	52
4.7 pav. Pašalinių koordinatų pavyzdys	52
4.8 pav. Sustojimo paklaida priklausomai nuo greičio	54
4.9 pav. Roboto greičio mažinimo procesas	55
4.10 pav. H-Bridge tipo variklių valdiklio krypčių schema.....	55
4.11 pav. Išsiskleidžiantis paėmimo įrenginys.....	56
4.12 pav. Išsiskleidžiančio paėmimo padėkliukas	57

4.13 pav. Magnetinis paėmimo įrenginys	58
4.14 pav. Magnetinio paėmimo įrenginio padėkliukas prekei	58
4.15 pav. Vakuuminis paėmimo įrenginys.....	59
5.1 pav. Tyrimo Nr.1 apibendrinti rezultatai.....	61
5.2 pav. Vidutinė koordinačių paklaida pagal skirtingus metodus	61
5.3 pav. Skirtingų metodų rezultatų palyginimas su realiais duomenimis.....	62
5.4 pav. Nuvažiuoto atstumo diagrama kiekvieno bandymo metu	63

Santrumpų ir terminų sąrašas

AGV (angl. *Automated guided vehicle*) – automatizuoto vedžiojimo transporto mašina .

AMR (angl. *Autonomous mobile robot*) – autonominis mobilus robotas.

SLAM (angl. *Simultaneous Localization and Mapping*) – kompiuterizuotas nežinomos aplinkos žemėlapių sudarymas arba atnaujinimas naudojant aplinkos jutiklių informaciją

RFID (angl. *Radio-frequency identification*) – tai objekto ar objektų žymėjimo ir sekimo *sistema*, kuri pagrįsta radijo dažnio signalo panaudojimu objekto žymėjime esančios informacijos įrašymui ir nuskaitymui.

FROG (angl. *Free ranging on grid*) – autonominio judėjimo būdas, kai įrenginys vadovaujasi magnetais ant pagrindo įmontuotais magnetais, kad orientuotis ir judėti erdvėje.

UML – (angl. *Unified Modeling Language*) unifikauta modeliavimo kalba.

Mikrokompiuteris – nedidelių dydžių (telpa ant paprasto stalo) kompiuteriai, kurių centrinio procesoriaus darbą atlieka mikroprocesorius

Mikrokontroleris – plačios(arba specializuotos) paskirties programuojama ir mikroprocesorių turinti schema

NFC (angl. *Near-field communication*) – artimojo lauko komunikacijos technologija

USB (angl. *Universal serial bus*) – universalioji jungtis naudojama kompiuteriuose

WIFI – belaidžio ryšio technologija. Wi-Fi belaidžio ryšio technologija leidžia realizuoti duomenų perdavimo tinklus panaudojant plačiajuostį radijo ryšį

API (angl. *Application Programming Interface*) – tai sąsaja, kurią suteikia kompiuterinė sistema, biblioteka ar programa tam, kad programuotojas per kitą programą galėtų pasiekti jos funkcionalumą ar apsikeistų su ja duomenimis.

1. Įvadas

Robotai ir mašinos jau ne vieną dešimtmetį padeda žmonėms automatizuoti procesus bei palengvinti darbo krūvį įvairiose gyvenimo srityse: gamyboje, žemės ūkyje, medicinoje, karo pramonėje ir kitur [1]. Tačiau dvidešimt pirmajame amžiuje robotų projektavimo ir išsivystymo lygis stipriai išaugo bei žengė didelį žingsnį aplinkos suvokimo srityje, o tai leido sukurti tobulesnius – automatizuoto vedžiojimo ir visiškai autonominius robotus.

Šie robotai gali matyti aplinką, analizuoti, girdėti, komunikuoti tarpusavyje ir net užmegzti pokalbį su žmogumi. Vis labiau į žmonių kasdienybę įsibraunantys robotai sugeba atlikti namų ruošos darbus, tvarkyti aplinką, žaisti žaidimus prieš žmogų, atlikti sudėtingas medicinos operacijas ir netgi būti panašūs ir elgtis kaip tikri žmonės [2]. Tačiau pagrindinė problema yra ta, kad robotas nėra žmogus. Lyginant su paprastais robotais, automatizuotas arba autonominis robotas negali būti visada tiksliai suprogramuotas, kad atliktų tik iš anksto nustatytus veiksmus. Savarankiškas robotas turi sugebėti dirbti dinamiškoje aplinkoje, reaguoti į besikeičiančias sąlygas ir, svarbiausia, pats priimti sprendimus. Būtent tai ir yra didžiausias pastarųjų metų mokslo iššūkis, kuriuo siekiama roboto intelektą priartinti prie žmogaus, kad būtų padidintos aplinkos pažinimo bei sąveikos galimybės. Tam reikia tobulinti tiek techninės (sensoriai, varikliai ir pan.), tiek ir programinės įrangos sritis.

Robotai vis plačiau pritaikomi ir logistikos srityje. Robotizacija leidžia automatizuoti procesus, kuriuos seniau galėdavo atlikti tik žmonės. Daiktų perkėlimas iš vienos vietos į kitą, inventorizacijos atlikimas, automatinis užsakymų vykdymas ir kt. Tai tik keli dalykai, kuriuos robotai leidžia atlikti logistikos srityje taip mažinant darbų kaštus ir laiką.

Šiame darbe bus bandoma įgyvendinti vieną iš svarbiausių automatizuoto sandėlio sistemai skirtų komponentų – automatiškai gebantį judėti robotą, parašyti jo valdymo sistemą bei atlikti roboto tikslumo bei tinkamumo tyrimą.

1.1. Dokumento paskirtis

Šiame dokumente aprašoma sistemos kūrimo veiksmai ir rezultatai. Pirmiausia analizuojamos jau egzistuojančios panašios sistemos bei technologijos, aptariami veikimo principai ir pasirenkamos technologijos. Toliau pateikiami projekto reikalavimai, aprašoma jau sukurta sistema bei gauti rezultatai. Tyrimo dalyje analizuojamos sukurtos sistemos problemos, realizuojami patobulinimai, kurie išbandomi eksperimentinėje dalyje.

1.2. Projekto tikslas

Tikslas – suprojektuoti logistikai skirtą roboto prototipą, kuris sugebėtų automatiškai judėti uždaroje sandėlio patalpoje, gebėtų perkelti daiktus iš vienos vietos į kitą ir atlikti kitas su logistika susijusias funkcijas. Robotų funkcijų ir užduočių valdymui taip pat turi būti sukurta robotų sistemos valdymo programinė įranga. Ištirti veikimo tikslumą.

Kuriamos sistemos naudotojas būtų įmonės, kurios turi sandėlius ir nori automatizuoti logistikos veiklą. Viena iš siekiamybių – sukurti sistemą prieinamą tiek vidutinėms, tiek mažoms įmonėms ir kuri pilnai automatizuotų sandėlio darbą, o šis robotas būtų viena iš sudedamųjų sistemos dalių.

1.3. Uždaviniai

1. Išanalizuoti technologijas ir rinkoje egzistuojančius sprendimus;
2. Suprojektuoti roboto prototipą, kuris turėtų reikiamus komponentus logistikos funkcijoms atlikti;
3. Suprogramuoti automatizuoto roboto važiavimo logiką ir misijų bei komponentų valdymo programinę įrangą;
4. Ištestuoti programinę ir techninę įrangą bei veikimo tikslumą;
5. Iširti sukurtos sistemos panaudojamumo galimybes, nustatyti problemas ir pasiūlyti patobulinimus.

1.4. Užsakovas

Šio projekto užsakovo organizacija yra uždaroji akcinė bendrovė – „Aksmo sistemos“. Tai startuolis, kuris užsiima programinės įrangos kūrimu logistikos sprendimams. Ši kuriama sistema yra įmonės verslo idėja, kuria siekiama sukurti produktą sandėliui automatizuoti.

Sukurtos programinės įrangos autorinės teisės priklausys UAB „Aksmo sistemos“. Produktą bus siekiama vystyti ir parduoti kaip paslaugą plačiajai rinkai. Potencialūs sistemos pirkėjai yra įmonės, kurios turi sandėlius ar parduotuves ir nori automatizuoti jų veiklą. Siekiamybė sukurti produktą, kuris pagal kainą būtų prieinamas ne vien didelėms, bet ir vidutinėms bei mažoms įmonėms.

2. Analitinė dalis

2.1. Analizės tikslas

Analizės metu bus nagrinėjama automatizuotų robotų tipai ir jų esminiai projektavimo bei pritaikymo aspektai, kad būtų nustatyta kokias technologijas reikia panaudoti projektuojant šiame darbe kuriamą logistikos roboto prototipą ir kuriant jo programinę įrangą. Taip pat gilinsimės į pasaulyje naudojamus komponentus, kurie robotams padeda orientuotis erdvėje ir atlikti judėjimo operacijas. Atlikus analizę bus galima priimti sprendimą kokio tipo robotas ir sistema turi būti realizuota, kad tenkintų norimus dalykinės srities reikalavimus bei kainos ribas.

2.2. Robotų tipai

Kaip savo straipsnyje „The Difference Between AGVs and Mobile Robots“ teigia Matt'as Edward'as [?], nėra vieno apibrėžimo, kas yra mobilusis robotas. Tai dažniausia apibūdinama, kaip savarankiškas robotas, kuris gali nukeliauti iš vienos vietos į kitą, kad atliktų tam tikras užduotis. Kalbant abstrakčiai, tai net ir paprastas dulkių siurblys, kuris važinėja po jūsų namus, yra mobilusis robotas, tačiau dabar technologijos daug patobulėjo, ir mobilieji robotai buvo išskirstyti į dvi grupes, kurios susiję su autonomiškumu. Šis skirtumas atsirado ne dėl pačių robotų skirtumo, bet dėl to, kad kiekvienas iš šių tipų skirtingais metodais atlieka savo užduotis.

Taigi autonominiam judėjimui skirti robotai yra skirstomi į dvi rūšis: automatinio vedžiojimo robotai – toliau vadinami AGV (angl. Automated guided vehicle) ir autonominiai mobilieji robotai. AGV tipo robotai ilgą laiką buvo pagrindiniai ir vieninteliai, kurie buvo naudojami autonominiam robotizuotų įrenginių judėjimui, tačiau vis labiau sudėtingėjanti darbo aplinka ir nuolat besikeičiantys reikalavimai privertė ieškoti kitokių sprendimų. Taip mobilieji robotai buvo pritaikyti dirbti autonomiškai ir tapo autonominiais mobiliaisiais robotais – toliau AMR.

Žemiau apžvelgsime šių dviejų tipų skirtumus ir veikimo principus.

2.2.1. Automatizuoto vedžiojimo transporto priemonės

Automatizuoto vedžiojimo priemonės (angl. automated guided vehicle) yra nepilotuojamos priemonės, kurios daugiausia naudojamos daiktams, mažiems ar dideliems, iš vienos vietos į kitą uždarse patalpose perkelti. Ši technologija veikia principu iš taško į tašką (angl. point-to-point) ir yra labai paplitusi gamybos ir logistikos srityse [6]. Nors naujausios bevielės technologijos leidžia AGV įrenginiams judėti pasitelkiant lazerio technologijas [14], tačiau pagrindinis AGV veikimo principas yra fiksuoto kelio sekimas – įrenginys turi sekti jam iš anksto nutiestą fiksuotą kelią [12].



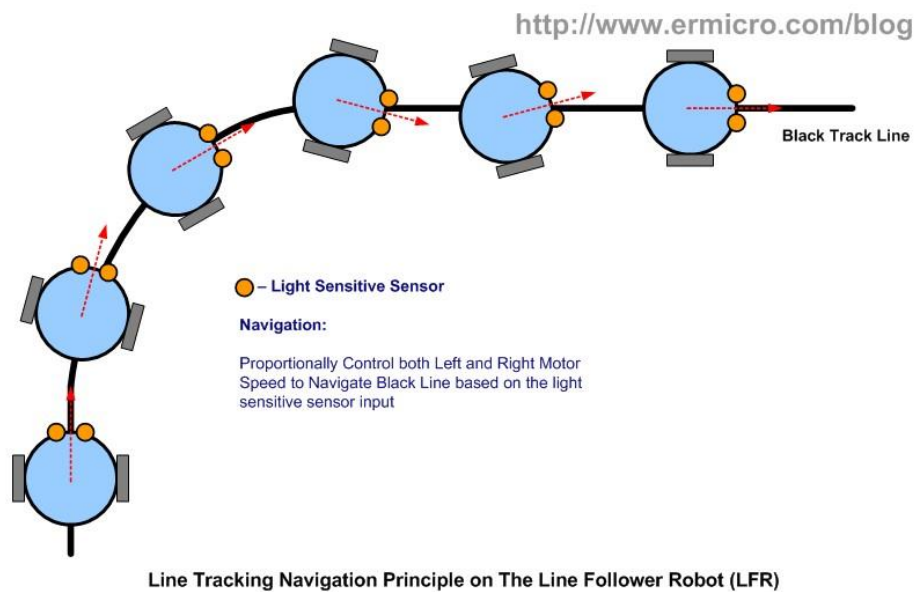
2.1 pav. Robotams nutiesto kelio pavyzdys [7]

Pozicionavimui AGV sistemose dažniausiai naudojami šie sprendimai [5]:

- magnetinės juostos (angl. *Magnetic tape*) [13]
- atspindinčios šviesą juostos (angl. *Reflective tape*)
- lazerinė trianguliacija (angl. *Lazer triangulation*) [15]
- magnetiniai taškai (angl. *Magnetic spot*) [15]
- švyturiai (angl. *Beacon*) [7]

Šiuo atveju keliai arba įranga jiems sukurti yra fiksuoti ir poreikis modifikuoti kelią reikalauja sustabdyti visą sistemą, kad fiziškai būtų galima pakeisti roboto kelio išdėstymą. Kita vertus, sekimo užduotis lengvai realizuojama, nes paprasčiausiu atveju robotui reikia turėti tik vieną jutiklį, kuris sektų nubrėžtą kelią (žr. 2.2 pav.).

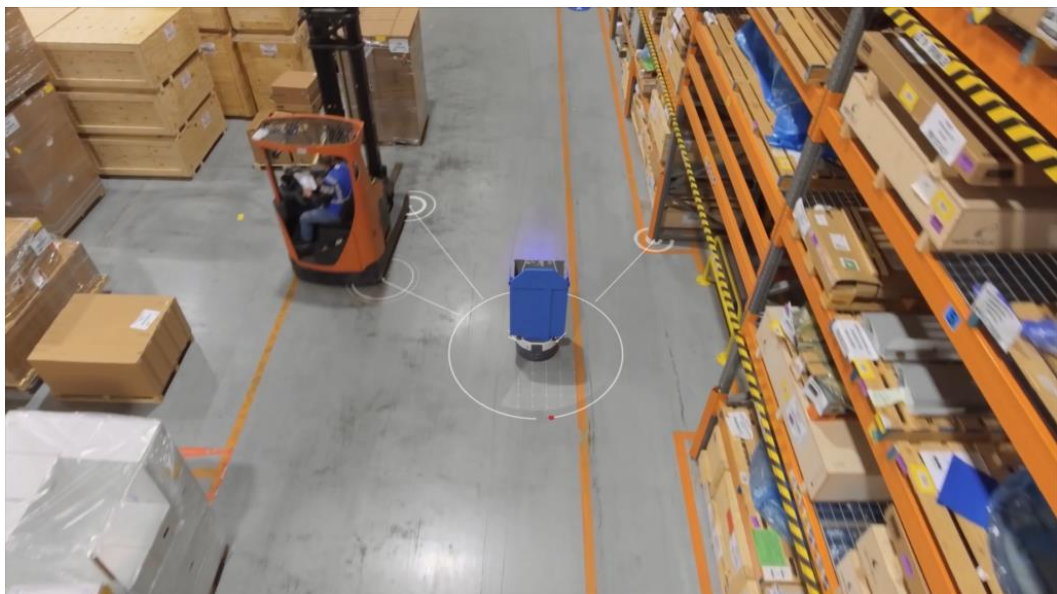
Vengiant kliūčių AGV įrenginiai naudoja atstumo, lazerinius ar kitokio tipo jutiklius, kurie automatiškai sustabdo įrenginį, jei jo kelyje atsiranda kliūtis, tačiau įrenginys negali judėti toliau ir pats priimti sprendimų, kol kliūtis nepasitraukia, tai yra didelis skirtumas lyginant su autonominiu įrenginiu [7].



2.2 pav. Fiksuotas vienos krypties roboto kelias

2.2.2. Autonominis mobilusis robotas

Autonominis robotas atlieka tą pačią užduotį, važiuoja iš taško A į tašką B. Tačiau būdas, kaip jis tą daro, yra visiškai skirtingas. Vietoj fiksuotos kelio nurodymo infrastruktūros (laidai, juostos, magnetai ir kt.) autonominis robotas naudoja tik aplinką skenuojančius jutiklius [3]. Tokie įrenginiai turi dirbtinį intelektą, kuris atsižvelgiant į susidariusią nenuspėjamą situaciją gali priimti naujus sprendimus ir mokytis [7].

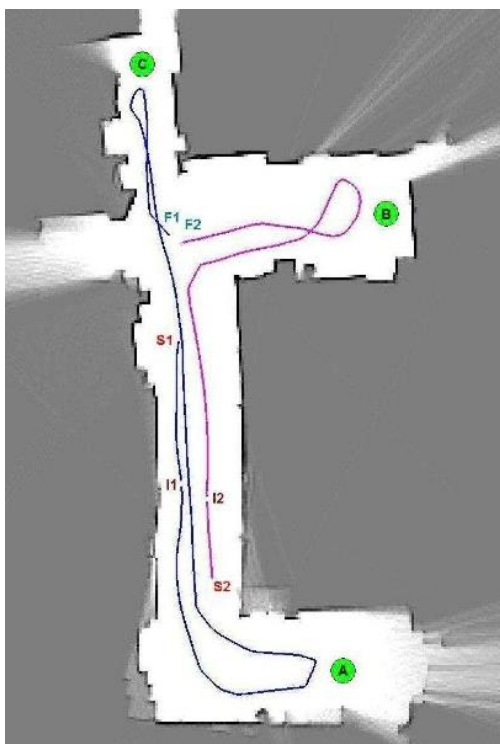


2.3 pav. AMR tipo robotas sandėlio aplinkoje [10]

AMR tipo robotuose dažniausiai naudojami šie metodai [18]:

- SLAM algoritmai;
- aplinkos skenavimas.

Pagrindinis šiuo metu judėjimui patalpų viduje taikomas metodas yra pagrįstas aplinkos skenavimu lazeriu ir judėjimo planavimo algoritmais, kurie leidžia nepriklausomai nuo aplinkos pasikeitimų dinamiškai priimti sprendimus taip padidinant efektyvumą ir prisitaikymą prie naujų sąlygų. Šiam tikslui taip pat yra naudojami SLAM algoritmai. Paprastai pirmojo paleidimo metu robotas valdant žmogui apvažiuoja darbo vietos erdves, jutikliai surenka informaciją ir sudaro aplinkos žemėlapi, pagal kurį vėliau robotas gali judėti erdvėje.



2.4 pav. SLAM metodu sudarytas aplinkos žemėlapis [11]

2.2.3. Robotų tipų palyginimas

Žemiau pateiktoje lentelėje aprašome pagrindinius AGV ir AMR robotų skirtumus ir aspektus.

1 lentelė. Robotų tipų palyginimas

Kriterijus/Tipas	AGV	AMR
Judėjimas aplinkoje	Naudojama pasirinkta infrastruktūra: magnetinės juostos, barkodai, RFID ir kt.	Autonominis judėjimas, realiu laiku apžvelgiama aplinka ir priimami sprendimai dinamiškoje aplinkoje.
Kliūčių vengimas	Įrenginys sustoja prie kliūčių ir nejuda kol jos nepašalinamos	Sustoja prie kliūties ir gali ją apvažiuoti, perskaičiuojant maršrutą.
Lankstumas naujoje aplinkoje	Reikalingas infrastruktūros suplanavimas ir įdiegimas	Lengvai sukuriamas aplinkos žemėlapis žvalgant naują patalpą
Išplečiamumas	Galima pridėti naujus kelius papildant esamą infrastruktūra	Viskas valdoma programiškai koreguojant programinę įrangą ir apžvelgiant naują patalpą.
Intelektas	Robotas gali judėti tik iškarto žinomu keliu ir atlikti apibrėžtas užduotis	Realiu laiku priimami sprendimai leidžia prisitaikyti prie besikeičiančios aplinkos

Darbo greitis	Darbo greitis mažesnis, nes važiuoja tik nustatytu keliu	Didesnis darbo greitis, nes visada paskaičiuojamas artimiausias maršrutas
Krovimas	Krovimo stotelė	Krovimo stotelė
Realizacija	Lengvesnė	Sudėtingesnė

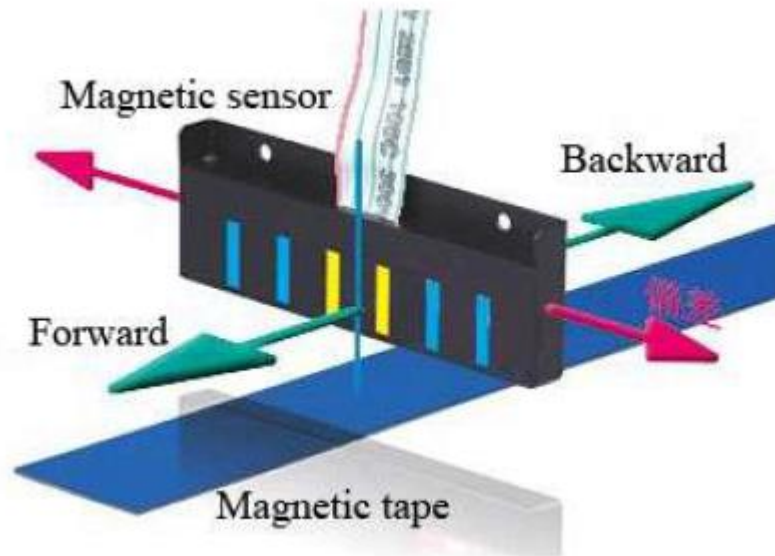
Apžvelgus palyginimų lentelę (žr. 1 lentelė), galime daryti išvadą, kad AMR tipo robotai yra tobulesnė AGV tipo versija. Jie lengviau pritaikomi naujose aplinkose, nereikalauja infrastruktūros keitimo, turi dirbtinį intelektą, kuris gali realiu laiku reaguoti į sąlygų pasikeitimus bei dažniausiai pateikia didesnį darbo našumą, nes paskaičiuojamas geriausias kelionės maršrutas, o ne važiuojama tik nustatytu keliu. Verta paminėti, kad AGV tipo robotų realizacija turėtų būti lengvesnė, nes apima mažiau aplinkos faktorių bei nenaudoja dirbtinio intelekto, kuris reikalauja daug sąnaudų, jog išvystytų sėkmingą ir stabilią sistemą.

2.3. AGV judėjimo metodai

2.3.1. Magnetinė juosta

Magnetinių juostų principu paremtas judėjimas yra populiarus, nes lengvai įgyvendinamas ir reikalauja mažai išlaidų [8]. Magnetinės juostos metodas pagrįstas ant kelio nutiesta juosta, kuri savo viduryje generuoja magnetinį lauką, o ant važiuojančio įtaiso esantys magnetiniai jutikliai aptinka skleidžiamą lauką ir taip nustato įrenginio poziciją linijos atžvilgiu [9, 13]. Šis magnetinės linijos metodas naudoja idėją, kuri yra atėjusi iš įterpiamų laidų grindyse metodo, tačiau šis linijų metodas yra daug patogesnis, nes linijas užtenka klijuoti ant grindų ir nereikia keisti pagrindo struktūros, kaip būdavo įterpiamų laidų metode. Taigi šis orientavimo metodas laikomas viduriu tarp įrenginio kainos ir sistemos lankstumo ypač kalbant apie sistemas, kurios turi ribotą biudžetą kuriant AGV sistemas [12]. Magnetinės navigacijos modeliai dažnai yra papildomi jutikliais, kurie geba nustatyti papildomų žymių formą ar siunčiamus signalus, kad būtų nustatyti papildomi supančios aplinkos duomenys.

Principinė magnetinės juostos sekimo fiksuotam keliui vizualizacija matoma 2.5 paveikslėlyje. Magnetinę juostelę sudaro derva, sumaišyta su kietuoju anglies tirpalu. Magnetinių dalelių pasirinkimas yra labai svarbus, nes jis yra tiesiogiai susijęs juostos sukeliamu magnetinio lauko stipriu [13]. Atsižvelgiant į jutiklio tipą ant kiekvieno iš jų yra skirtingas skaičius matavimo taškų, kurie gali užfiksuoti magnetinio lauko signalus. Dėl šios priežasties kiekvienas jutiklis gali turėti skirtingą maksimalų matavimo plotį ir šoninio nuokrypio dydį, kol jutiklis visiškai nenukrypo nuo linijos. Pavyzdžiui, 2.5 paveikslėlyje vaizduojamas jutiklis turi 6 matavimo taškus.

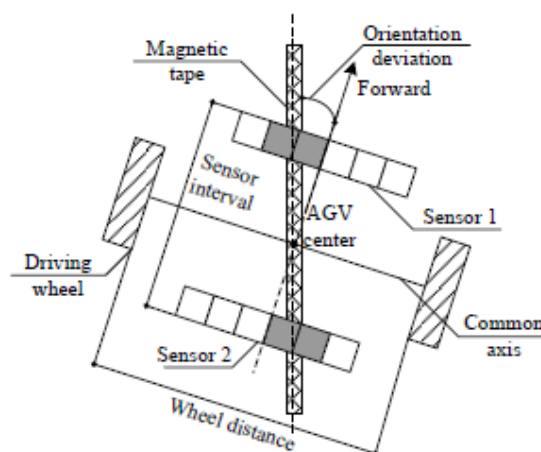


2.5 pav. Magnetinio tipo sekimas AGV įrenginiui [12]

Kaip savo darbe [12] teigia Xing Wu ir kolegos, vieno sensoriaus, kad tiksliai būtų nustatytas nuokrypis, neužtenka. Dėl šios priežasties, kaip vaizduojama 2.6 paveikslėlyje, robotuose įdiegiami du magnetiniai jutikliai – priekyje ir gale, kurie kartu matuoja nuokrypius nuo magnetinės juostos. Turint omenyje, kad jutiklių vidurio taškas atitinka roboto vidurio tašką tarp ašių, šoninis nuokrypis gali būti išreiškiamas tokia formule:

$$e_d = \frac{e_{d1} + e_{d2}}{2} \quad (1)$$

Čia e_{d1} – priekinio jutiklio rodmenys, e_{d2} – galinio jutiklio duomenys.



2.6 pav. Nukrypimo nuo linijos matavimas [12]

Orientacinis roboto nuokrypis gali būti išreikštas tokia formule:

$$e_\theta = \tan^{-1} \frac{e_{d1} - e_{d2}}{w_s} \quad (2)$$

Čia e_{d1} – priekinio jutiklio rodmenys, e_{d2} - galinio jutiklio duomenys, W_s – atstumo intervalas tarp dviejų sensorių.

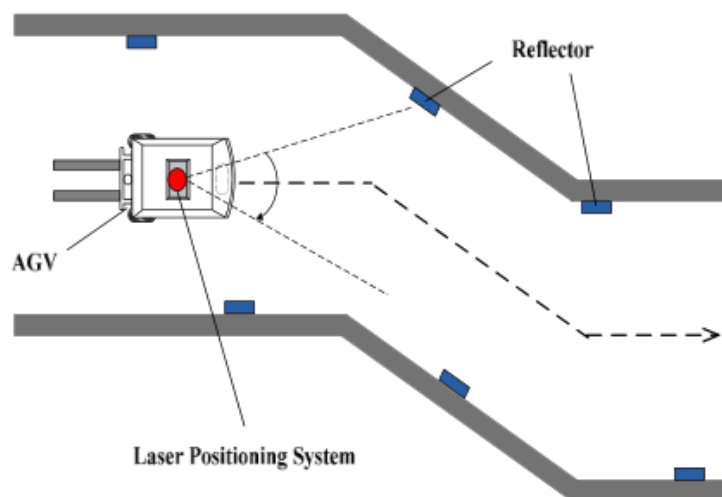
Turint tokius duomenis, galima kurti algoritmą variklių greičiui reguliuoti, kad būtų atliekami posūkiai ar išlaikoma tiesi roboto pozicija.

2.3.2. Šviesą atspindinčios juostos

Šio metodo principas labai panašus į prieš tai minėtų magnetinių juostų. Įrenginys pasinaudodamas infraraudonųjų spindulių jutikliais arba kompiuterine rega seka ant grindų nupieštas arba kitaip pridėtas linijas arba kitokias žymes. Robotas seka šias linijas taip žinodamas savo trajektoriją. Kai kuriuose taikymuose galima sutikti ir kitokių sprendimų, kai tam tikrame aukštyje išdėstomi tam tikros formos objektai-žymės, kuriuos robotas jutikliais stebi arba liečia ir taip nustato savo padėtį bei žino savo padėtį [17].

2.3.3. Lazerinė trianguliacija

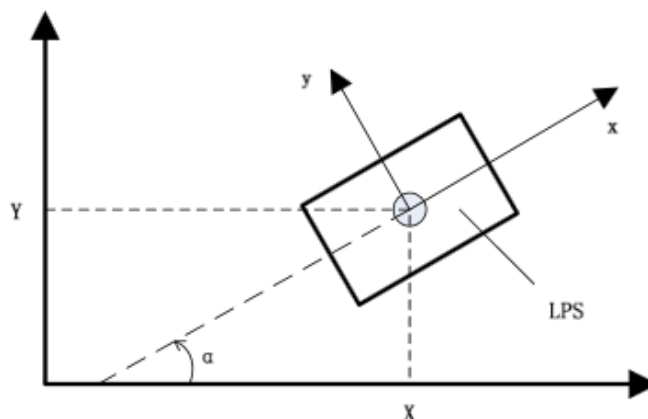
Trianguliacija yra pozicijos nustatymo metodas, paremtas atstumo iki žinomų objektų matavimu. Tokiose sistemose dažniausia yra trys arba daugiau siūstuvų, sumontuotų nustatytose vietose ir vienas robote. Šis pozicionavimo metodas atsirado norint išspręsti įterptiniais laidais ir magnetinių juostų paremto judėjimo problemas. Kaip savo darbe [16] teigia Kun Fan ir kolegos šis metodas suteikia privalumus tokius kaip: geresnis tikslumas, didesnis greitis, realiu laiku vykdomas kliūčių vengimas. Šiame metode dažniausiai naudojamas lazerinis įrenginys sudarytas iš lazerio ir jutiklio, kuris skenuoja aplinką 360 laipsnių kampu [15]. Lazerinio pozicionavimo sistema paremti įrenginiai matuoja laiką per kurį šviesa paleista iš lazerio atsimuša į tam tikrą pagrindą ir sugrįžta atgal (žr. 2.7 pav.) [16].



2.7 pav. Įrenginio pozicijos nustatymas erdvėje pagal lazerio atspindžius [16]

Lazerinio pozicionavimo sistema susideda iš dviejų koordinačių sistemų: reliatyviosios ir absoliučiosios. Reliatyvioji nusako įrenginio vietą erdvėje priklausomai nuo iš važiuoklės kodavimo įrenginio (angl. *shaft encoder*) gaunamų duomenų, tačiau ji gali turėti nemažas paklaidas. Kad būtų sumažintos klaidų tikimybės, sistema pasitelkia absoliučiąsias koordinates, kurios patikslina įrenginio vietą ir kryptį bei mažina netikslumų tikimybę. Koordinatės gaunamos, priklausomai nuo

atspindinčių įrenginių žinomos padėties, kurie yra išdėlioti roboto judėjimo erdvėje [16]. Žemiau esančiame 2.8 paveikslėlyje matomos absoliutinės įrenginio koordinatės x ir y judėjimo erdvėje.



2.8 pav. Absoliutinių ir vietinių koordinatinių sistema vietos nustatymui [16]

2.3.4. Švyturiai

Dar viena AGV robotų lokalsios vietos nustatymo koncepcija susijusi su švyturių (angl. beacons) naudojimu. Metodo esmė – šaltinių, skleidžiančių radijo signalus, išdėstymas roboto veikimo zonoje. Švyturių sistemose taikomi trilateracijos ir trianguliacijos principai [18]. Švyturiai lengvai aptinkami ir teikia informaciją apie įrenginio poziciją. Šioje pozicionavimo sistemoje svarbus tikslus švyturių išdėliojimas aplinkoje [19]. Tokios sistemos užtikrina didelį patikimumą, tačiau joms reikalingos didelės montavimo ir priežiūros išlaidos [18].

2.4. Autonominio judėjimo metodai

2.4.1. SLAM algoritmai

Autonominio judėjimo robotai naudoja papildomą įrangą, tokią kaip orientyrai [20,21] ar inerciniai įrenginiai [22]. Tačiau norint įgalinti robotą suprasti jį supančią aplinką reikia naudoti algoritmus pagrįstus dvimačių ar trimačių aplinkos žemėlapių sudarymu [23].

Kaip savo darbe [18] teigia Olga Fiodorova metodo esmė žemėlapių sudarymu pagrįstas padėties nustatymas. Tai metodikos, kuriose mobilusis robotas naudoja jutiklius supančios aplinkos žemėlapiui sukurti. Šis vietinis žemėlapis gali būti sulygintas su turimu žemėlapiu jau saugomu atmintyje ir jei nustatomas atitikmuo – robotas gali apskaičiuoti savo esamą padėtį. Dažniausiai yra naudojami arba iš anksto programine įranga paruošti žemėlapiai, arba robotas pats juos sukuria judėdamas erdvėje.

Dažniausiai naudojami algoritmai:

- L-SLAM – nuskaitymas skeneriu [24]
- LSD-SLAM [25], ORB-SLAM [26], VSLAM [27] – kompiuterinė rega
- DV-SLAM [28] – magnetiniai jutikliai
- Ir kt.

2.4.2. Aplinkos skenavimas

Dar viena metodologija, kuri suteikia galimybę robotui suprasti jį supančią aplinką tai pastovus aplinkos skenavimas tam skirtu įrenginiu, pavyzdžiui skeneriu. Lazeriniu skeneriu gauti aplinkos duomenys dažniausiai naudojami statinėms bei dinaminėms kliūtims nustatyti [29]. Kaip savo disertacijoje [18] teigia jau minėta Olga Fiodorova skenerio teikiamą informaciją galima panaudoti ir patalpos profiliui sudaryti, kuris savo ruožtu padėtų surasti mobiliojo roboto padėties koordinates.

Šiame metode naudojami lazeriniai skeneriai veikia jau prieš tai aptartu trianguliacijos principu. Lazerio spindulys projektuojamas ant skenuojamo objekto, o vaizdo kamera, kuri yra pritaisyta šone lazerio, fiksuoja lazerio spindulius ant objektų [18].

Pagrindinis skenavimo privalumas – šis metodas pakankamai greitas, bekontaktis ir tikslus. Tačiau gali kilti problemų jei lazerio spindulys susiduria su tokia aplinka kaip: šviečiantys, skaidrūs ar šviesą sugeriantys paviršiai.

2.5. Metodų palyginimas

2 lentelė. Metodų paklaidų palyginimas

Metodas	Paklaida	Atstumas
Magnetinė juosta	Nežinoma vieta	Patalpa
Šviesą atspindinčios juostos	Nežinoma vieta	Patalpa
SLAM	Nuo 1mm iki 1dm	Patalpa
Švyturiai	Apie 7cm	5m
Žemėlapių sudarymas (SLAM)	Tarp 10cm ir keleto metrų	Patalpa
Aplinkos skenavimas	Nuo 3mm iki 5mm	Patalpa

Apibendrinant lentelę galime teigti, kad seniausi ir paprasčiausi metodai judėjimui ir pozicijos nustatymui yra magnetinės ir šviesą atspindinčios juostos. Šie metodai nesuteikia informacijos apie roboto poziciją erdvėje. Tam šie metodai turi būti apjungti su papildomomis žymomis(RFID) ar panašiomis technologijomis.

Egzistuoja tobulesni metodai tokie kaip SLAM arba švyturiai, kurie iš karto suteikia tam tikro tikslumo informaciją apie tai kur konkrečiu laiko momentu yra robotas. Tačiau šie metodai reikalauja brangesnės techninės įrangos, galingesnių skaičiavimo resursų ir žinių specifinėje srityje.

2.6. Analizės išvados

Analizės metu aptarti mobiliųjų robotų tipai ir navigavimo bei pozicijos nustatymo metodai uždaroje erdvėje. Gilinantį į kitų autorių darbus pastebėta, kad roboto judėjimas uždaroje erdvėje ir jo padėties nustatymas yra sudėtingas procesas. Daugelis egzistuojančių metodų naudoja papildomą įrangą (kamos, girokopai, žymės, orientyrai). Vis labiau populiarėja metodai paremti aplinkos skenavimu, kurie leidžia pažinti aplinką be papildomos fizinės aparatūrinės įrangos.

Vertinant kuriamos sistemos poreikį, tai yra, kad nereikalingas realaus laiko kelių perskaičiavimas ir sandėliuose keliai gali būti fiksuoti, galima daryti prielaidą, kad geriausias pasirinkimas bus bandyti realizuoti sistemą pritaikant linijos sekimo technologiją apjungiant ją su komponentais, kurie padėtų nustatyti roboto padėtį aplinkoje.

3. Projektinė dalis

3.1. Sistemos paskirtis

Kliento idėja, sukurti automatizuotą sandėlio sistemą, kurioje nereikėtų žmogaus įsikišimo atliekant prekių sandėliavimo, paėmimo, perkėlimo, inventorizacijos ir panašias užduotis. Pagrindinis sistemos tikslas sėkmingai nuvažiuoti iš vieno taško į kitą uždaroje sandėlio erdvėje. Taip pat sistema turi sustoti atsiradus kliūtims kelyje. Kaip papildomas funkcionalumas sistemoje taip pat bus kuriamas išorinio paėmimo įrenginio valdymo modulis bei automatinio lentynų dydžio skenavimo modulis. Visas šias užduotis turi sugebėti atlikti sukurta misijų valdymo programinė įranga. Kuriamas automatizuotas robotas yra viena iš sudedamųjų automatizuoto sandėlio sistemos dalių, kuri atneštų judėjimo funkcionalumą aplinkoje.

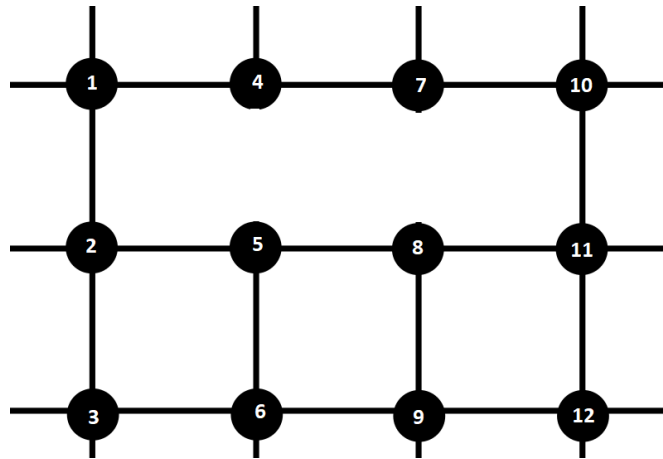
3.2. Projekto kūrimo pagrindimas

Projekto pagrindimui galime pasiremti dabartinio sukurto prototipo problemomis. Šiuo metu yra sukurtas fiziškai prie sandėlio lentynų tvirtinimas įrenginys, kuris gali paimti prekes, tačiau esama sistema negali būti lengvai plečiama ir neturi galimybės judėti sandėlio erdvėje, nes kiekviena sandėlyje esanti prekių lentyna turi fiziškai prie jos primontuotą mechaninį prekių perkėlimo mechanizmą. Tai sukuria dideles sistemos plečiamumo problemas, neleidžia lengvai pridėti papildomų lentynų, lengvai pakeisti išplanavimo bei svarbiausia neleidžia prekės perkelti iš vienos fizinės vietos į kitą. Pavyzdžiui paimti prekę ir perkelti ją į kitame sandėlio gale esančią lentyną ar nuvežti prekės į atsiėmimo punktą. Šios problemos privertė ieškoti sprendimo, kuris pagerintų kuriamos sistemos plečiamumą ir atneštų judėjimo funkcionalumą aplinkoje.

3.3. Pasirinktas sprendimas

Atlikus technologijų analizę ir įvertinus keliamus funkcinius bei nefunkcinius reikalavimus nuspręsta kurti sistemą susidedančią iš:

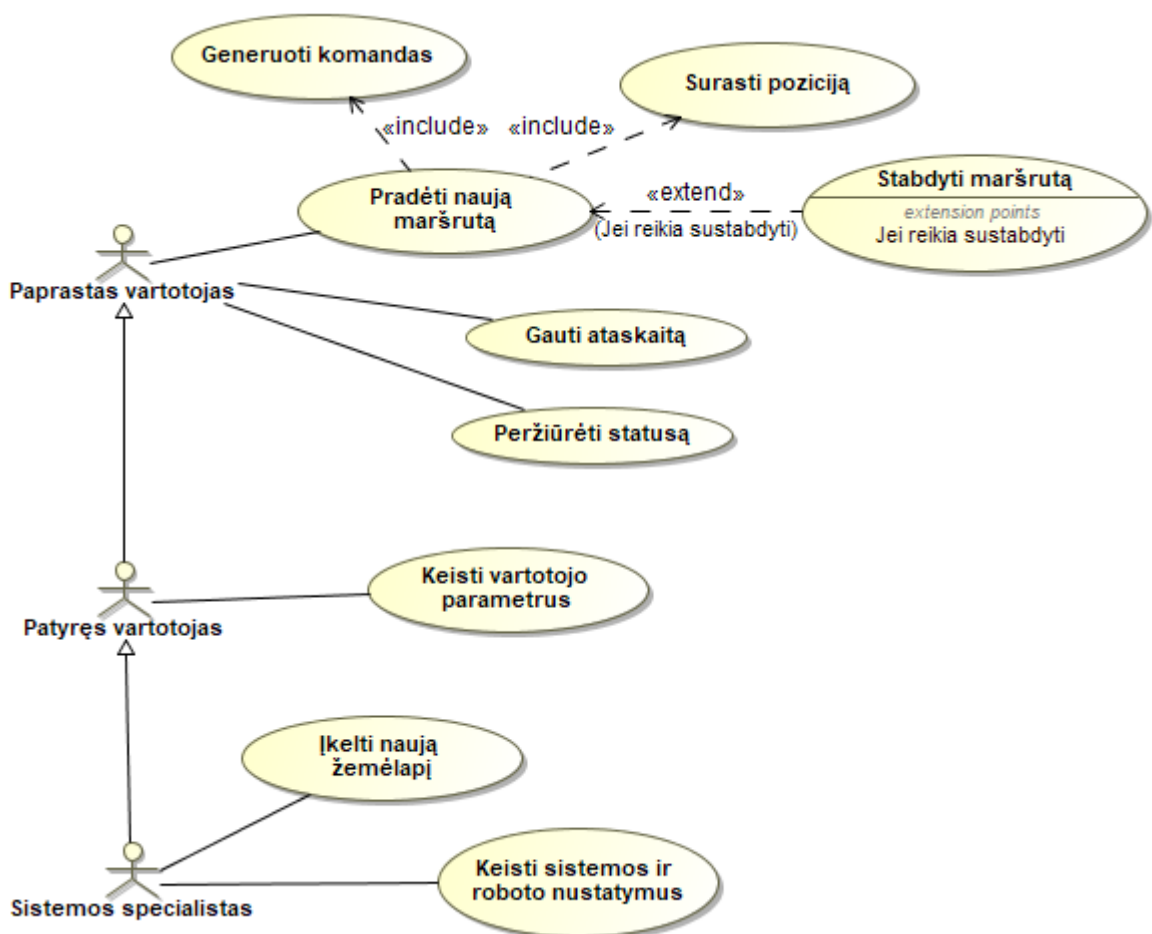
- AGV (angl. automated guided vehicle) tipo roboto, kuris nėra pilnai autonominis, o orientuojasi pagal tam tikrą iš anksto paruoštą infrastruktūrą.
- Robotui ant žemės bus nubrėžtos baltos arba juodos spalvos linijos, kurias jis turės sekti naudojant infraraudonųjų spindulių jutiklius.
- Pozicijos nustatymui naudojamos tam tikruose linijos taškuose pritvirtintos RFID žymės.
- Judėjimas vyks ir programinė įranga bus kuriama orientavimuisi tinklinio (angl. grid) tipo žemėlapyje (žr. 3.1 pav.).
- Bus kuriamos dvi atskiros valdymo sistemos: roboto realaus laiko mechaninių elementų valdymo sistema ir misijų valdymo sistema
- Sistemos tarpusavyje ir su kitais komponentais komunikuos per sugeneruotas komandas.



3.1 pav. Tinklinio tipo žemėlapio pavyzdys su numeriais pažymėtomis RFID žymėmis

3.4. Sistemos panaudos atvejai

Šiame skyrelyje naudojant UML notaciją pavaizduota sistemos panaudos atvejų diagrama. Taip pat lentelėse žemiau aprašytas kiekvienas panaudos atvejis atskirai.



3.2 pav. Roboto valdymo sistemos panaudos atvejų diagrama

3 lentelė. Panaudos atvejo „Pradėti naują maršrutą aprašas“

PA 1. Pradėti naują maršrutą	
Tikslas/uždavinys. Įvesti naują maršrutą, kur robotas turi nuvažiuoti.	
Aprašymas. Vykdamas šį PA, sistema bandys suskaičiuoti artimiausią kelią iki reikiamos vietos ir jei toks yra pradės vykdyti PA „Generuoti komandas“ kur vykdomas komandų generavimas.	
Prieš-sąlyga	-
Aktorius	Paprastas vartotojas, patyręs vartotojas, sistemos specialistas
Sužadinimo sąlyga	Norima, kad robotas nuvažiuotų į tam tikrą vietą
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuoti PA
Pagrindinis scenarijus	
Pagrindinis įvykių srautas	Sistemos reakcija
1. Naudotojas įveda naujo maršruto komandą ir nurodo tikslą, kur reikia važiuoti.	1.1. Sistema suskaičiuoja artimiausią maršrutą ir pradeda generuoti komandas. Sugeneravus komandas prasideda komandų vykdymas.
2. Baigiamas PA.	
Po-sąlyga	Suskaičiuojamas maršrutas, sugeneruojamos komandos ir pradeda važiavimo procedūra.
Alternatyvūs scenarijai	
1. Naudotojas įveda naujo maršruto komandą ir nurodo tikslą, kur reikia važiuoti.	1.1 Sistema neranda maršruto ar iškyla kita problema. Išvedamas klaidos pranešimas
2. Baigiamas PA	

4 lentelė. Panaudos atvejo „Stabdyti maršrutą“ aprašas

PA 2. Stabdyti maršrutą	
Tikslas/uždavinys. Sustabdyti maršrutą ir robotą.	
Aprašymas. Vykdamas šį PA, sistema sustabdys vykdomą maršrutą ir robotą.	
Prieš-sąlyga	PA 1. Pradėti naują maršrutą
Aktorius	Paprastas vartotojas, patyręs vartotojas, sistemos specialistas
Sužadinimo sąlyga	Norima, kad sistema sustabdytų vykdomą maršrutą ir robotą.
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuoti PA
Pagrindinis scenarijus	
Pagrindinis įvykių srautas	Sistemos reakcija
1. Naudotojas įveda stabdymo komandą	1.1. Jei maršrutas vykdomas, sistema sustabdo maršrutą.
2. Baigiamas PA.	
Po-sąlyga	Suskaičiuojamas maršrutas ir pradeda važiavimo procedūra.
Alternatyvūs scenarijai	
1. Naudotojas įveda stabdymo komandą važiuoti.	1.1 Jei maršrutas nevykdomas, išvedamas klaidos pranešimas.
2. Baigiamas PA	

5 lentelė. Panaudos atvejo „Gauti ataskaitą“ aprašas

PA 3.Gauti ataskaitą	
Tikslas/uždavinys. Gauti roboto darbo ataskaitą.	
Aprašymas. Vykdam šį PA, sistema sugeneruos darbo ataskaitą	
Prieš-sąlyga	
Aktorius	Paprastas vartotojas, patyręs vartotojas, sistemos specialistas
Sužadinimo sąlyga	Norima , kad sistema sugeneruotų ataskaitą.
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuoti PA
Pagrindinis scenarijus	
Pagrindinis įvykių srautas	Sistemos reakcija
1. Naudotojas įveda ataskaitos generavimo komandą	1.1. Sistema gražina statistinius duomenis.
2. Baigiamas PA.	
Po-sąlyga	Sugeneruotas ir gražintas statistikos rinkinys.

6 lentelė. Panaudos atvejo „Peržiūrėti statusą“ aprašas

PA 4.Peržiūrėti statusą	
Tikslas/uždavinys. Peržiūrėti roboto statusą.	
Aprašymas. Vykdam šį PA, sistema parodys statusą , kas šiuo metu vykdoma.	
Prieš-sąlyga	
Aktorius	Paprastas vartotojas, patyręs vartotojas, sistemos specialistas
Sužadinimo sąlyga	Norima peržiūrėti statusą
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuoti PA
Pagrindinis scenarijus	
Pagrindinis įvykių srautas	Sistemos reakcija
1. Naudotojas įveda statuso komandą	1.1. Sistema gražina šiuo metu esantį statusą ir kitą informaciją
2. Baigiamas PA.	
Po-sąlyga	Sugeneruota ir vartotojui gražinta statuso informacija

7 lentelė. Panaudos atvejo „Keisti vartotojo parametrus“ aprašas

PA 5.Keisti vartotojo parametrus	
Tikslas/uždavinys. Pakeisti nustatymus, kurie yra prieinami paprastam vartotojui	
Aprašymas. Vykdam šį PA, sistemai nusiunčiami nauji vartotojo nustatymai.	
Prieš-sąlyga	
Aktorius	Patyręs vartotojas
Sužadinimo sąlyga	Keisti nustatymus
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuoti PA
Pagrindinis scenarijus	
Pagrindinis įvykių srautas	Sistemos reakcija
1. Naudotojas įveda statuso komandą ir parametrus keitimui	1.1. Gražinamas pakeitimo arba klaidos pranešimas

2. Baigiamas PA.	
Po-sąlyga	Pakeisti nustatymai

8 lentelė. Panaudos atvejo „Įkelti naują žemėlapi“ aprašas

PA 6.Įkelti naują žemėlapi	
Tikslas/uždavinys. Įkelti naują žemėlapi, kuriuo turi naviguoti robotas.	
Aprašymas. Vykiant šį PA, sistemoje įkeliamas naujas žemėlapis: gretimumo matrica ir dvimatė kryptių matrica, kurios pagalba robotas naviguoja aplinkoje.	
Prieš-sąlyga	
Aktorius	Sistemos specialistas
Sužadinimo sąlyga	Atnaujinti žemėlapio duomenis sistemoje
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuoti PA
Pagrindinis scenarijus	
Pagrindinis įvykių srautas	Sistemos reakcija
1. Naudotojas įveda žemėlapio keitimo komandą ir parametrus keitimui.	1.1. Sistema patikrina ar įkeliami duomenys teisingi ir jei teisingi išsaugo bei grąžina patvirtinimą.
2. Baigiamas PA.	
Po-sąlyga	Atnaujintas žemėlapis

9 lentelė. Panaudos atvejo „Keisti sistemos ir roboto nustatymus“ aprašas

PA 7.Keisti sistemos ir roboto nustatymus	
Tikslas/uždavinys. Pakeisti sisteminius ir roboto nustatymus, kurie yra prieinami tik patyrusiam vartotojui	
Aprašymas. Vykiant šį PA, sistemai nusiunčiami nauji sisteminiai ir roboto nustatymai.	
Prieš-sąlyga	
Aktorius	Patyręs vartotojas
Sužadinimo sąlyga	Keisti nustatymus
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuoti PA
Pagrindinis scenarijus	
Pagrindinis įvykių srautas	Sistemos reakcija
1. Naudotojas įveda statuso komandą ir parametrus keitimui	1.1. Grąžinamas pakeitimo arba klaidos pranešimas
2. Baigiamas PA.	
Po-sąlyga	Pakeisti nustatymai

10 lentelė. Panaudos atvejo „Surasti poziciją“ aprašas

PA 8. Surasti poziciją	
Tikslas/uždavinys. Surasti roboto poziciją erdvėje orientuojantis pagal žymes	
Aprašymas. Vykiant šį PA, sistema bandys surasti savo buvimo vietą. Tai yra robotas pradės judėti, kol suras RFID žymę, kad užfiksuotų savo buvimo vietą erdvėje	
Prieš-sąlyga	Robotas nežino savo buvimo vietos
Aktorius	Paprastas vartotojas, patyręs vartotojas, sistemos specialistas
Sužadinimo sąlyga	Norima kad robotas nustatytų savo vietą erdvėje
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA

Specializuoti PA	
Pagrindinis scenarijus	
Pagrindinis įvykių srautas	Sistemos reakcija
1. Naudotojas įveda komandą, kad pradėti ieškoti pozicijos.	1.1. Robotas pradeda judėti linija kol užfiksuoja RFID žymę ir sustoja.
2. Baigiamas PA.	
Po-sąlyga	Robotas žino savo vietą erdvėje

3.5. Funkciniai reikalavimai

Papildomos sistemos funkcijos, kurios turėjo būti realizuotos siekiant sėkmingo roboto funkcionavimo.

11 lentelė. Funkcinis reikalavimas FR-1

Reikalavimas #:	FR-1	Reikalavimo tipas:	V9	PA #:
Aprašymas:	Robotas turi saugiai judėti aplinkoje ir sustoti prieš priekyje esančias kliūtis			
Pagrindimas:	Robotas yra sunkus įrenginys, jis judės autonomiškai todėl svarbu, kad neįvyktų susidūrimai su aplinkoje judančiais ar esančiais kitais objektais.			
Šaltinis:	Sistemos užsakovas			
Atitikimo kriterijus:	Roboto priekyje atsiradus kliūčiai, kuri yra arčiau nei 30cm iki roboto priekio, jis privalo sustoti ir nejudėti kol kliūtis nebus pašalinta.			
Užsakovo pasitenkinimas:	3	Užsakovo nepasitenkinimas:	5	
Prioritetas:	5	Konfliktai:	Nėra	
Istorija:	Sukurtas: 2018-04-12			

12 lentelė. Funkcinis reikalavimas FR-2

Reikalavimas #:	FR-2	Reikalavimo tipas:	V9	PA #:	1
Aprašymas:	Sistema turi paskaičiuoti artimiausią atstumą iki reikiamo kelionės tikslo				
Pagrindimas:	Robotui svarbu pasirinkti ekonomiškiausius važiavimo kelius, kad judėjimas būtų kuo sklandesnis ir greitesnis, taip taupant energiją ir didinant misijų atlikimo greitį.				
Šaltinis:	Sistemos užsakovas				
Atitikimo kriterijus:	Artimiausio kelio radimui naudojamas Dijkstra algoritmas, kuris skirtas trumpiausio kelio paieškai svoriniame grafe. Taip surandamas mažiausia atstumą turintis kelias iki tikslo.				
Užsakovo pasitenkinimas:	3	Užsakovo nepasitenkinimas:	4		
Prioritetas:	4	Konfliktai:	Nėra		
Istorija:	Sukurtas: 2018-04-12				

13 lentelė. Funkcinis reikalavimas FR-3

Reikalavimas #:	FR-3	Reikalavimo tipas:	V9	PA #:	5
Aprašymas:	Sistema pateikia detalius klaidų pranešimus.				
Pagrindimas:	Sutrikus automatiniam roboto darbui labai svarbu greitai identifikuoti problemą ir žinoti kokių veiksmų imtis.				
Šaltinis:	Sistemos užsakovas				

Atitikimo kriterijus:	Sistema turi detalius klaidų pranešimus , kuriuos siunčia įvykus nenumatytiems atvejams bei kartu su statuso pateikimu.		
Užsakovo pasitenkinimas:	3	Užsakovo nepasitenkinimas:	3
Prioritetas:	2	Konfliktai:	Nėra
Istorija:	Sukurtas: 2018-04-12		

14 lentelė. Funkcinis reikalavimas FR-4

Reikalavimas #:	FR-4	Reikalavimo tipas:	V9	PA #:	1
Aprašymas:	Robotas turi sugebėti komunikuoti su išoriniu paėmimo įrenginiu				
Pagrindimas:	Robotas nuvažiavęs į reikiamą vietą turi atlikti tam tikrą prekės paėmimo užduotį, kad šią užduotį atlikti turi būti realizuota išorinio įrenginio valdymas, kuris paremtas GRBL biblioteka .				
Šaltinis:	Sistemos užsakovas				
Atitikimo kriterijus:	Sistema valdo išorinį paėmimo įrenginį.				
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	5		
Prioritetas:	5	Konfliktai:	Nėra		
Istorija:	Sukurtas: 2018-08-12				

15 lentelė. Funkcinis reikalavimas FR-5

Reikalavimas #:	FR-5	Reikalavimo tipas:	V9	PA #:	1
Aprašymas:	Robotas turi sugebėti pajudėti tam tikrą centimetrų kiekį ir sustoti.				
Pagrindimas:	Kad robotas galėtų surasti tikslią jam reikalingą sustojimo vietą , reikia , kad jis gebėtų matuoti savo nuvažiuotą atstumą ir sustoti kai atstumas yra pasiektas.				
Šaltinis:	Sistemos užsakovas				
Atitikimo kriterijus:	Roboto sustojimo paklaida +- 1cm.				
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	4		
Prioritetas:	4	Konfliktai:	Nėra		
Istorija:	Sukurtas: 2018-08-12				

16 lentelė. Funkcinis reikalavimas FR-6

Reikalavimas #:	FR-6	Reikalavimo tipas:	V9	PA #:	1
Aprašymas:	Programinė įranga turi gebėti iš gautų lazerinio atstumo jutiklio duomenų apskaičiuoti lentynų paėmimo pozicijas				
Pagrindimas:	Norint lengviau prisitaikyti prie besikeičiančių sandėlio reikalavimų, lentynų dydžių ir praplėsti funkcionalumą siekiama padaryti automatinio lentynų skenavimo programą, kuri gebėtų su kuo mažesniu žmogaus įsikišimu pati nustatyti kokio dydžio yra lentynos ir kiek jų yra				
Šaltinis:	Sistemos užsakovas				
Atitikimo kriterijus:	Apskaičiuotos lentynos paėmimo pozicijos paklaida neturi viršyti 0,8cm.				
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	2		
Prioritetas:	3	Konfliktai:	Nėra		

Istorija:	Sukurtas: 2018-08-12
------------------	----------------------

3.6. Nefunkciniai reikalavimai

3.6.1. Reikalavimai panaudojamumui

17 lentelė. Reikalavimas panaudojamumui NR-1

Reikalavimas #:	NR-1	Reikalavimo tipas:	V11	PA #:
Aprašymas:	Aiškios, lengvai suprantamos roboto valdymo komandos, tikslūs bei aiškūs klaidų bei patvirtinimų pranešimai.			
Pagrindimas:	Svarbu, kad sistema būtų draugiška vartotojui – tai yra suteiktų lengvą ir suprantamą valdymą ir grįžtamąjį ryšį apie dabartinę sistemos būseną ir elgseną,			
Šaltinis:	Sistemos užsakovas			
Atitikimo kriterijus:	Pateiktoje apklausoje daugiau kaip 60% vartotojų teigia, kad valdymas paprastas ir aiškus			
Užsakovo pasitenkinimas:	2	Užsakovo nepasitenkinimas:	2	
Prioritetas:	3	Konfliktai:	Nėra	
Istorija:	Sukurtas: 2018-04-12			

3.6.2. Reikalavimai vykdymo charakteristikos

18 lentelė. Reikalavimas vykdymo charakteristikoms NR-2

Reikalavimas #:	NR-2	Reikalavimo tipas:	V12	PA #:
Aprašymas:	Kelio maršruto iki tikslo skaičiavimas neturi trukti ilgai ir per daug trukdyti darbą bei sukelti nepatogumus.			
Pagrindimas:	Kelio skaičiavimas viena pagrindinių sistemos dalių, kuri turi veikti tiksliai ir greitai. Tai leidžia taupyti roboto tarnavimo laiką važiuojant trumpiausiu keliu, tačiau pats skaičiavimo procesas taip pat negali būti ilgas, nes trikdys sistemos darbą.			
Šaltinis:	Sistemos užsakovas			
Atitikimo kriterijus:	Kelio suradimas sistemoje esant iki 500 tikslo taškų neturi trukti ilgiau kaip 10 sekundžių.			
Užsakovo pasitenkinimas:	4	Užsakovo nepasitenkinimas:	5	
Prioritetas:	5	Konfliktai:	Nėra	
Istorija:	Sukurtas: 2018-04-12			

3.6.3. Reikalavimai veikimo sąlygoms

- Įrenginys dirbs uždaroje patalpoje
- Įrenginys turi gebėti veikti prie įvairaus apšvietimo
- Įrenginys turi sugebėti sekti tiek tamsios, tiek ir šviesios spalvos liniją.

3.6.4. Reikalavimai sistemos priežiūrai

19 lentelė. Reikalavimas sistemos priežiūrai NR-3

Reikalavimas #:	NR-3	Reikalavimo tipas:	V14	PA #:	6
Aprašymas:	Sistema turi turėti galimybę pakeisti judėjimo kelių išplanavimą.				
Pagrindimas:	Pritaikant sistemą naujoje aplinkoje arba keičiant jau esamą aplinką keičiami ir judėjimo žemėlapiai, todėl sistema turi sugebėti priimti naujus išplanavimo duomenis, juos išsaugoti ir naudoti.				
Šaltinis:	Sistemos užsakovas				
Atitikimo kriterijus:	Po naujo išplanavimo įkėlimo, sistema tinkamai juda aplinkoje naudodama naujus kelius				
Užsakovo pasitenkinimas:	3	Užsakovo nepasitenkinimas:	4		
Prioritetas:	4	Konfliktai:	Nėra		
Istorija:	Sukurtas: 2018-04-12				

3.7. Sistemos kūrimo rizikos

20 lentelė. Sistemos kūrimo rizikos

Nr.	Rizika	Tikimybė	Poveikis
1	Programuotojas neturi reikiamos kvalifikacijos ir žinių	Vidutinė	Didelis
2	Reikalavimų pasikeitimai, kurie reikalauja didelių pokyčių	Žema	Labai didelis
3	Komandos narys suserga ir negali dirbti tam tikrą laiką tarpą.	Didelė	Mažas
4	Sukurta sistema ir prototipas neatitinka reikalavimų.	Žema	Labai didelis
5	Vėlyvas reikiamų komponentų pristatymas	Žema	Mažas
6	Baigėsi suplanuotas biudžetas	Labai žema	Labai didelis
7	Robotas sužaloja aplinkinius ar padaro kitokios žalos	Labai žema	Labai didelis
8	Projekto einamasis planas neatitinka realaus plano	Žema	Vidutinis
9	Bloga roboto valdymo programinės įrangos implementacija	Vidutinė	Didelis
10	Vadovas neturi pakankamai dalykinės srities įgūdžių	Didelė	Vidutinis

3.8. Sistemos architektūra

3.8.1. Architektūros pateikimas

Architektūros pateikimui bus naudojama UML kalba bei grafinio modeliavimo įrankis Magic Draw. Siekiant atvaizduoti norimus sistemos architektūros aspektus reikės šių UML diagramų:

- Panaudojimo atvejų vaizdas – panaudojimo atvejų diagrama
- Statinis vaizdas – klasių diagramos ir sistemos paketų diagrama
- Dinaminis vaizdas – būsena, sekų, veiklos diagramos
- Išdėstymo vaizdas – išdėstymo diagrama

3.8.2. Architektūros tikslai ir apribojimai

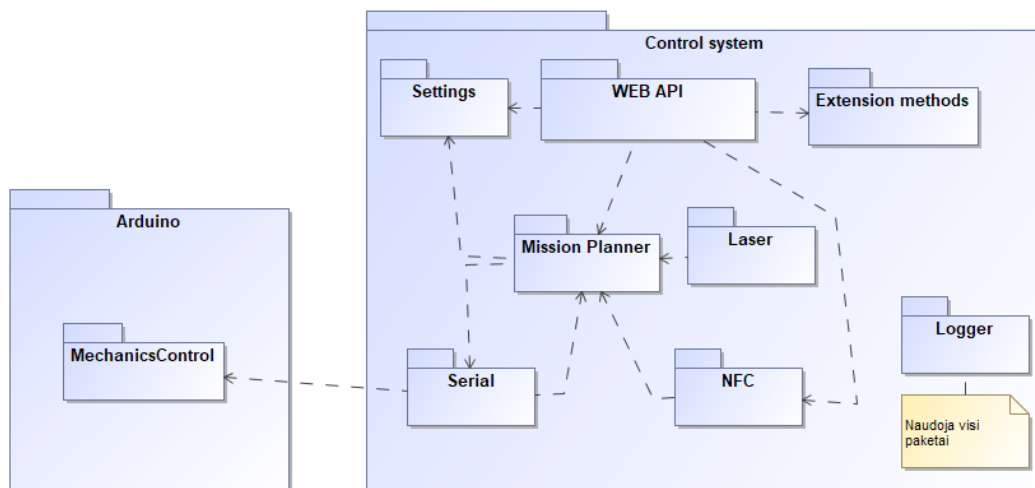
- Architektūros pateikimui bus naudojama UML kalba bei grafinio modeliavimo įrankis Magic Draw. Norint atvaizduoti norimus sistemos architektūros aspektus reikės šių UML diagramų:

- Panaudojimo atvejų vaizdas – panaudojimo atvejų diagrama
- Statinis vaizdas – klasių diagramos ir sistemos paketų diagrama
- Dinaminis vaizdas – būsenų, sekų, veiklos diagramos
- Išdėstymo vaizdas – išdėstymo diagrama

3.8.3. Sistemos statinis vaizdas

3.8.3.1. Apžvalga

Roboto maršruto planavimo sistema suskirstyta į 5 paketus, kurie pavaizduoti 3.3 pav.

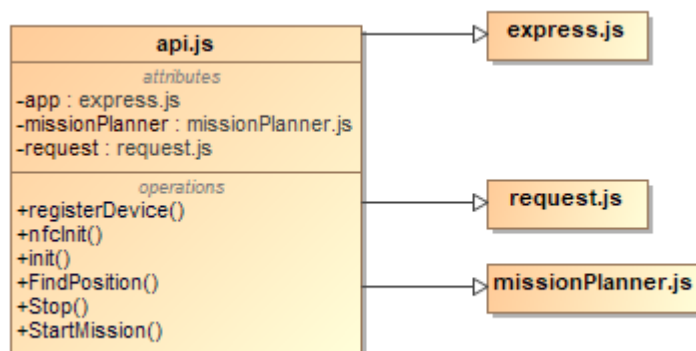


3.3 pav. Abstraktus sistemos skaidymas į paketus

3.8.3.2. Paketų detalizavimas

3.8.3.2.1. Paketas „WEB API“

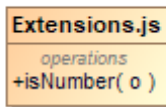
Šis paketas apima modulius, kurie skirti sistemos komunikavimui su išore. Šiame pakete yra ir pagrindinė sistemos klasė „api.js“, kuri startuoja programą.



3.4 pav. „WEB API“ paketas

3.8.3.2.2. Paketas „Extension methods“

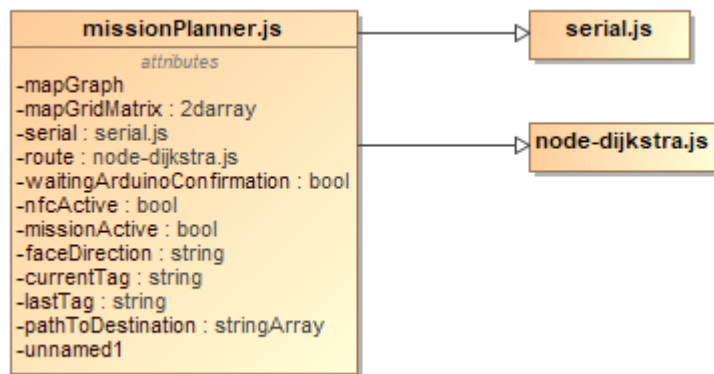
Paketas skirtas moduliams, kurie išplečia funkcionalumą paprastomis sistemos funkcionavimui reikiamomis funkcijomis.



3.5 pav. „Extensions“ paketas

3.8.3.2.3. Paketas „Mission planner“

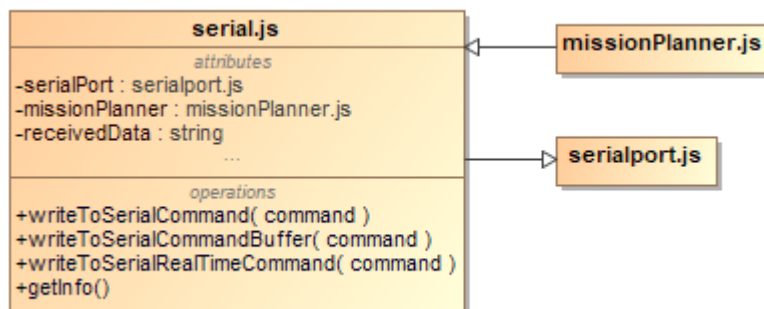
Paketas skirtas misijų planavimui ir vykdymui. Šiame pakete yra kelio skaičiavimo ir misijų vykdymo klasės.



3.6 pav. „Mission planner“ paketas

3.8.3.2.4. Paketas „Serial“

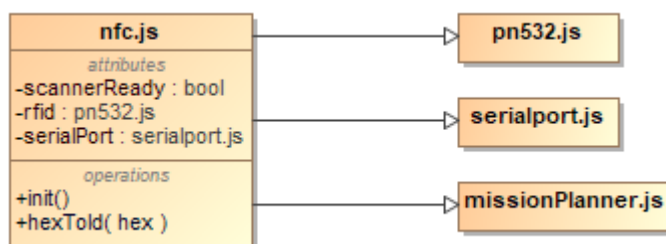
Paketas, kurio klasės skirtos komunikavimui su arduino įrenginiu per serial.



3.7 pav. „Serial“ paketas

3.8.3.2.5. Paketas „NFC“

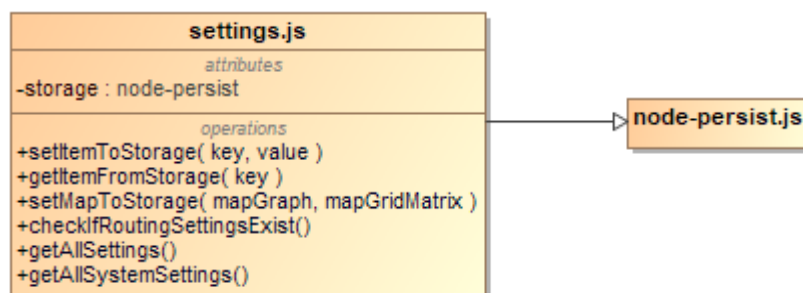
Paketas, kurio klasės susiję su NFC skaitytuvo funkcionavimu.



3.8 pav. „NFC“ paketas

3.8.3.2.6. Paketas „Settings“

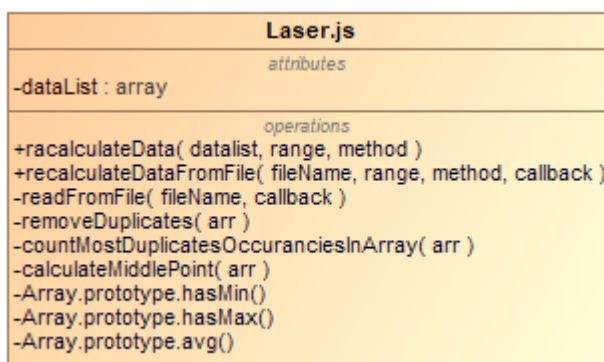
Paketas, kurio klasės susiję su sistemos nustatymų saugojimu ilgalaikėje aplinkoje bei gavimu iš jos.



3.9 pav. „Settings“ paketas

3.8.3.2.7. Paketas „Laser“

Paketas, kurio klasės susiję su lazerio valdymu ir duomenų apdorojimu. Žemiau vaizduojama tik duomenų apdorojimo paketo klasė.



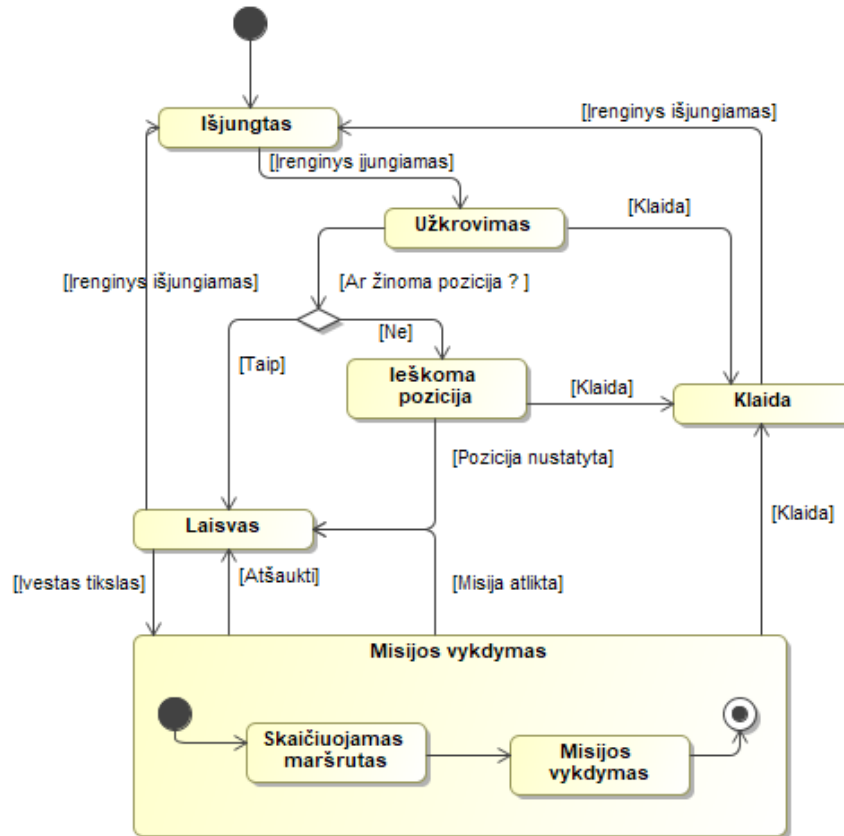
3.10 pav. „Laser“ paketas

3.8.4. Sistemos dinaminis vaizdas

Šiame skyriuje pateikiamos sekų, būsenų bei veiklos diagramos. Jos parodo bendrus kuriamos sistemos veiklos procesus ir leidžia geriau suprasti veiklos taisykles.

3.8.4.1. Būsenų diagramos

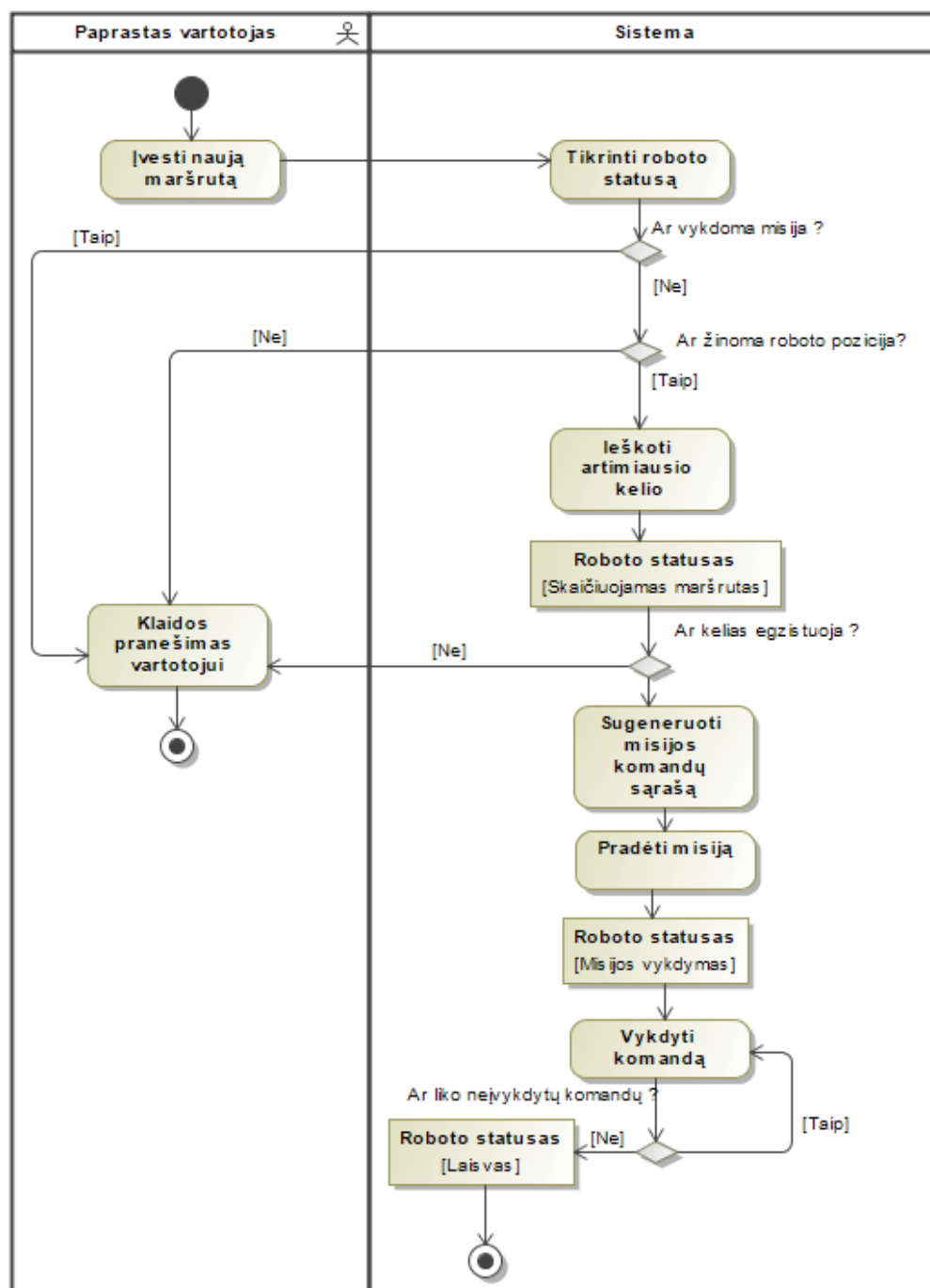
Būsenų diagramoje (žr. 3.11 pav.) vaizduoja pagrindinę įrenginio elgsena ir būsenas, kurias turi misijų planavimo ir valdymo sistema. Šios būsenos atspindi visą roboto situaciją ir būseną tam tikru atveju.



3.11 pav. Roboto valdymo sistemos būsenų diagrama

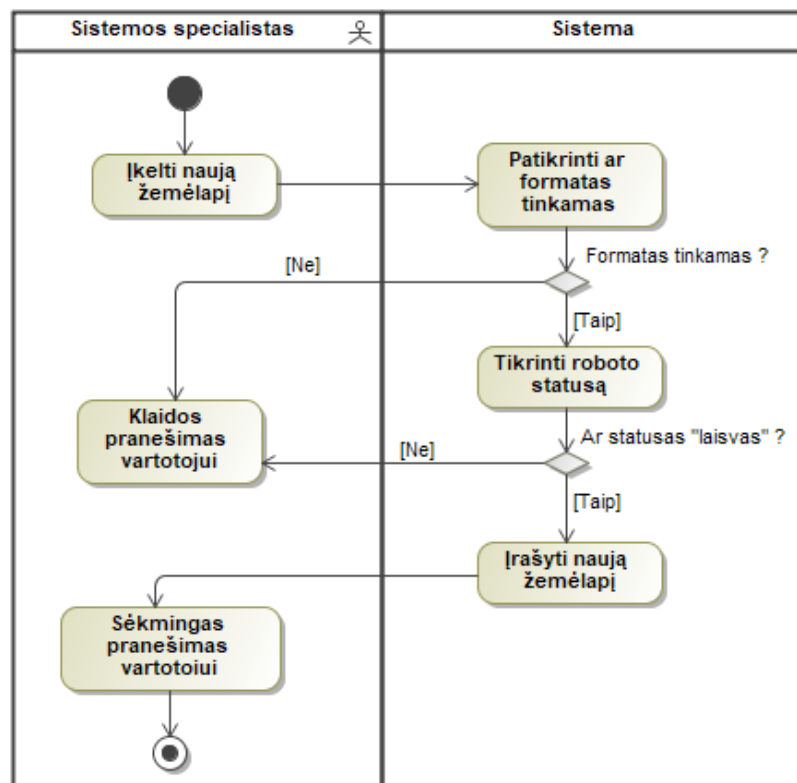
3.8.4.2. Veiklos diagramos

Veiklos diagramose pavaizduosime tris esminius procesus. Pats pagrindinis panaudos atvejis ir jo procesas kurį norėsim atvaizduoti yra „Pradėti naują maršrutą“ (žr. 3.12 pav.). Taip pat šiame skyriuje pateikiama panaudos atvejo „Įkelti naują žemėlapi“ veiklos diagrama (žr. 3.13 pav.) bei lazerio gautų duomenų apdorojimo procesas (žr. 3.14 pav.). Žemiau vaizduojama panaudos atvejo „Pradėti naują maršrutą“ veiklos diagrama. Diagramoje atsispindi kokios veiklos atliekas nuo vartotojo tikslo pateikimo iki užduoties pabaigos.



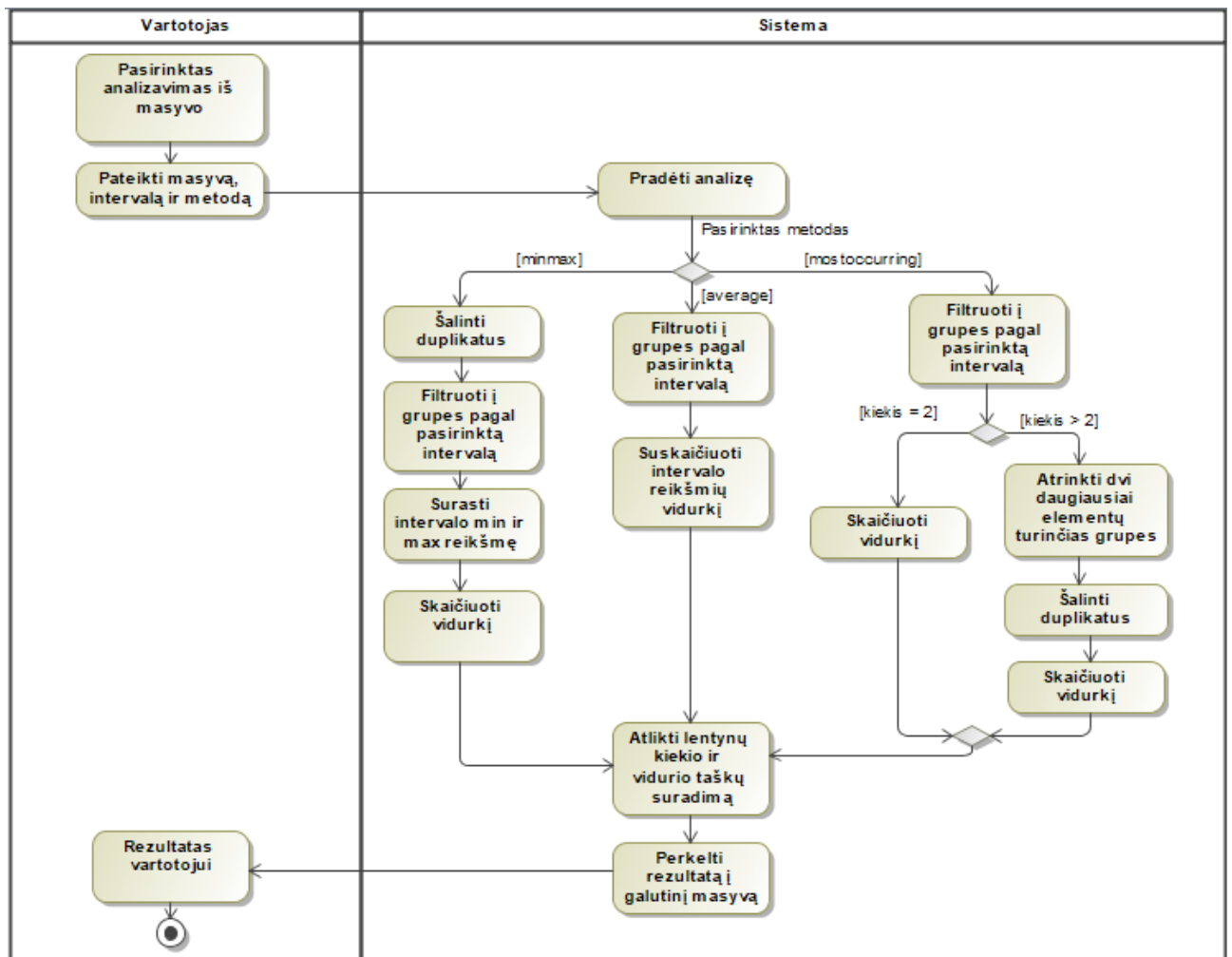
3.12 pav. Panaudos atvejo „Pradėti naują maršrutą“ veiklos diagrama

Žemiau pavaizduota ir kito panaudos atvejo „Įkelti naują žemėlapi“ veiklos diagrama, kurioje matoma pagrindinė įvykių seka įkeliant naują žemėlapi į robotų valdymo sistemą.



3.13 pav. Panaudos atvejo „Įkelti naują žemėlapi“ veiksmų diagrama

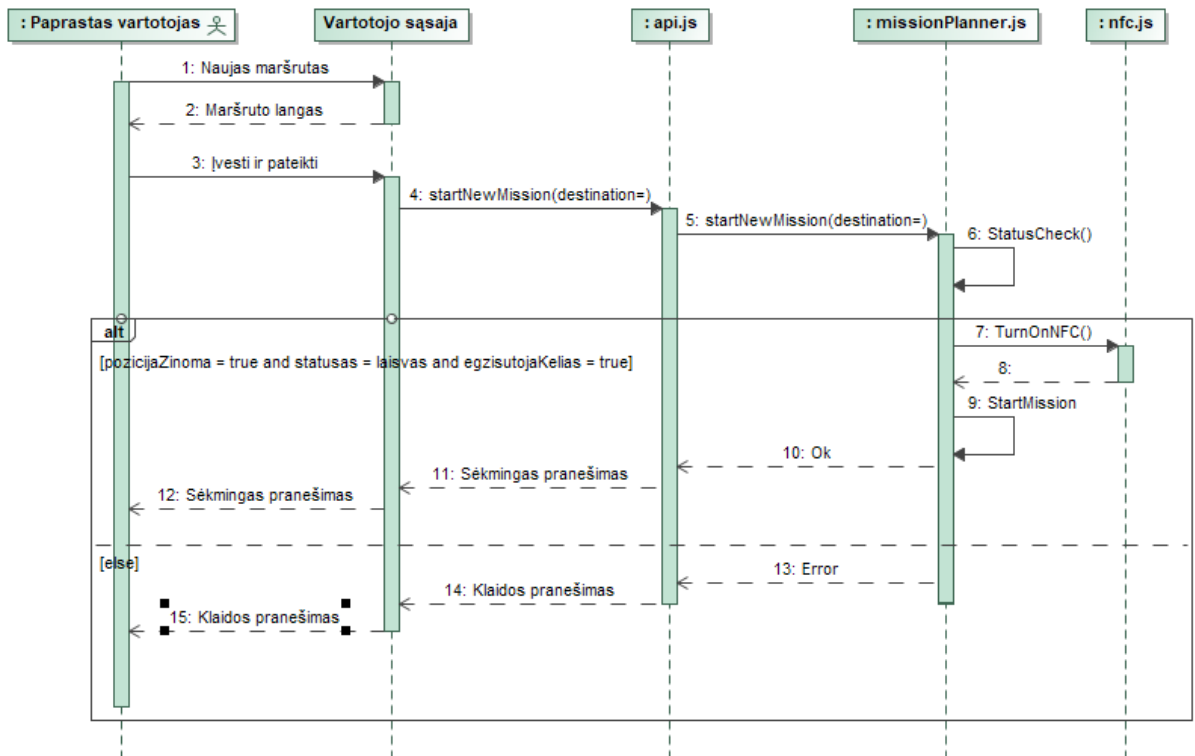
Paskutinė veiklos diagrama yra lentynų skenavimo metu gautų lazerio duomenų apdorojimas. Šis procesas apskaičiuoja lentynoje esančių celių centro koordinatės. Pirminėje programinės įrangos versijoje buvo realizuotas tik vienas algoritmas(diagramoje pavaizduotas „average“), o tyrimo ir eksperimentų metu pridėti dar du. Supaprastintas procesas pavaizduotas žemiau.



3.14 pav. Lazero duomenų apdorojimo proceso veiklos diagrama

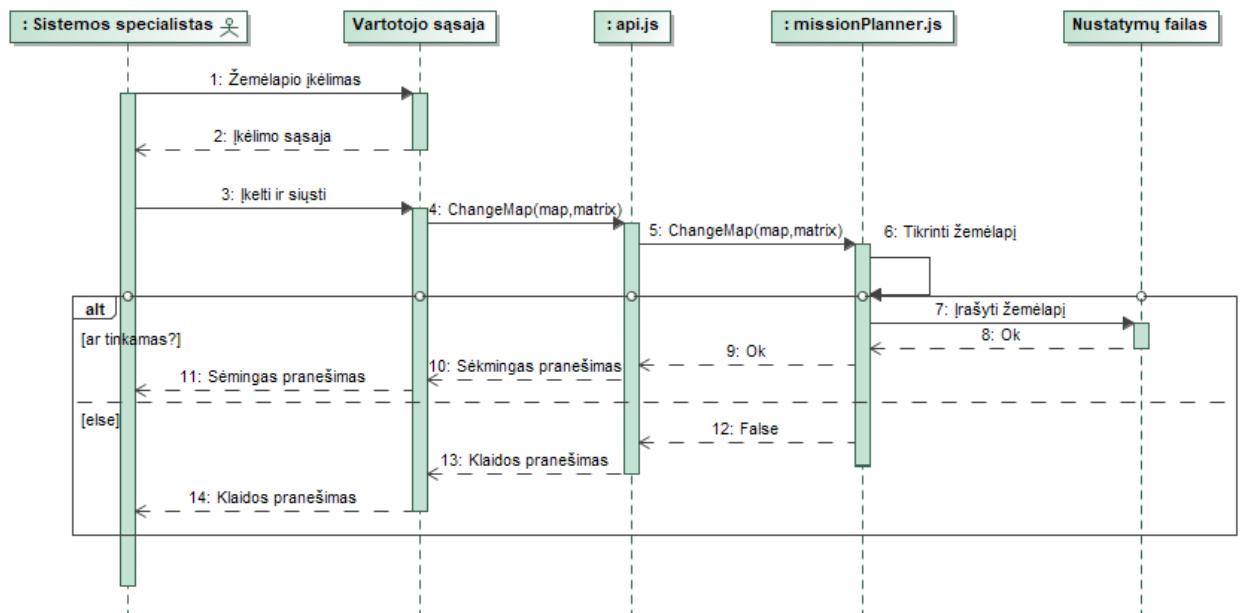
3.8.4.3. Sekų diagramos

Sekų diagramos pateikiamos dvi. Jos skirtos tiems patiems kaip ir ankstesniame poskyryje matytiems panaudos atvejams, nes jie reikalauja didesnio veiksmų eigos sudėtingumo. Žemiau šio teksto esantis paveikslėlis vaizduoja panaudos atvejo „Pradėti naują maršrutą“ sekų diagramą.



3.15 pav. Panaudos atvejo „Pradėti naują maršrutą“ sekų diagrama

Žemiau esančiame paveikslėlyje pavaizduotas žemėlapių keitimas valdymo sistemoje. Šis procesas leidžia sistemoje pakeisti turimą sandėlio kelių išplanavimą.



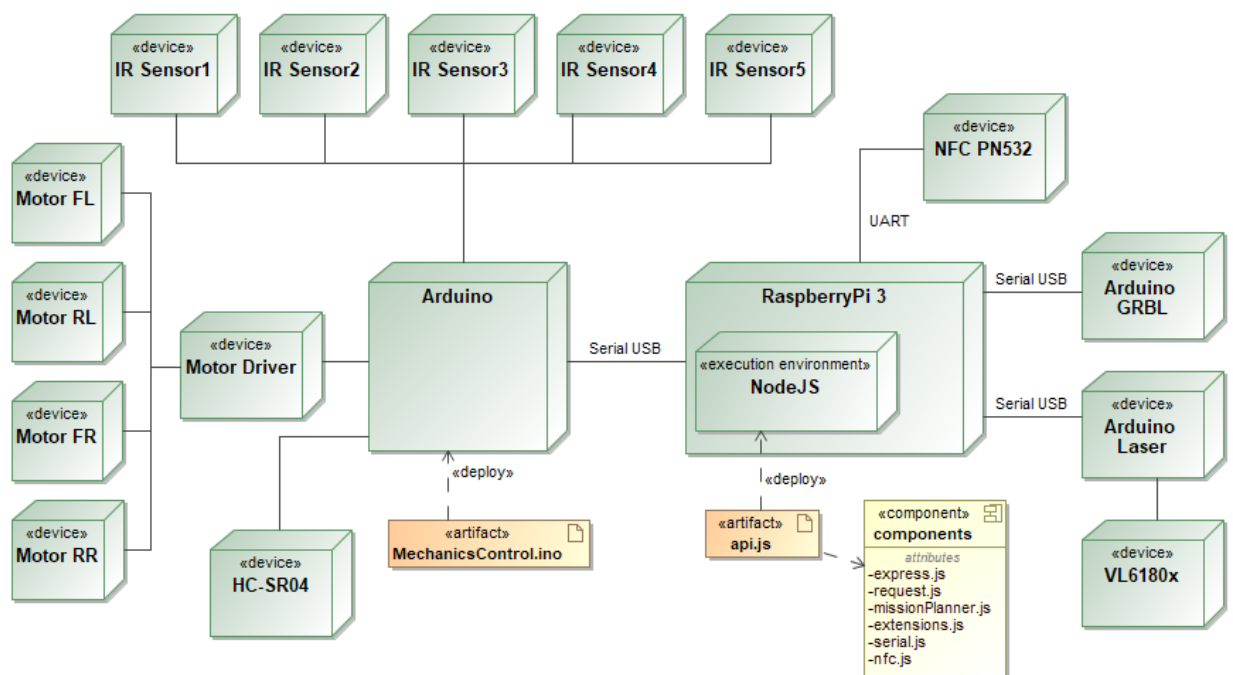
3.16 pav. Panaudos atvejo „Įkelti naują žemėlapią“ sekų diagrama

3.8.4.4. Išdėstymo vaizdas

Sistema susideda iš dviejų pagrindinių įrenginių: Arduino mikrovaldiklio ir RaspberryPi mikrokompiuterio.

Arduino mikrovaldiklis skirtas valdyti ir kontroliuoti realaus laiko įrenginius: roboto variklius, ultragarsinį atstumo jutiklį ir linijos sekimo šviesos jutiklius.

Kitas įrenginys RaspberryPi 3, kuriame įdiegta Linux operacinė sistema su NodeJS vykdymo aplinka skirtas kontroliuoti roboto darbą planuojant maršrutus ir valdant suplanuoto maršruto eigą (reguliuojant variklių darbą). Prie šio mikrokompiuterio taip pat yra prijungtas NFC skaitytuvas, kurio pagalba robotas orientuojasi erdvėje. Kompiuteris per USB serial prievadą komunikuoja su Arduino mikrovaldikliu, kuris valdo mechaninę ir realaus laiko dalį, specialiomis komandomis. Iš išorės komandas įrenginys gauna per integruotą WIFI ryšį. Taip pat prie RaspberryPi kompiuterio jungiami dar 2 papildomi mikrovaldikliai, kurie valdo išorinį paėmimo įrenginį (Arduino GRBL) ir lazerinį atstumo jutiklį (Arduino Laser). Visa tai galime matyti žemiau esančioje išdėstymo diagramoje (žr. 3.17 pav.).



3.17 pav. Roboto valdymo sistemos išdėstymo diagrama

3.8.4.5. Duomenų vaizdas

Roboto valdymo sistema nenaudos reliacinės duomenų bazės. Sistemoje naudojama gretimumo matrica su svoriais grafui atvaizduoti ir dvimatė kryptinių matrica, kuri nurodo kokios kryptimi susikirtimuose bus reikiama grafo viršūnė.

Gretimumo matrica susideda iš eilutėse surašytų viršūnių ir stulpeliuose išvardintų viršūnių su svoriais, su kuriomis ta viršūnė sujungta.

```

var mapGraph =
{
  '3951782987': { '2068019531': 1, '3952631627': 1 },
  '2068019531': { '3951782987': 1, '192251467': 1, '2068154187': 1 },
  '192251467': { '2068019531': 1, '1263818059': 1 },
  '1263818059': { '192251467': 1, '2068154187': 1, '195844427': 1 },
  '2068154187': { '2068019531': 1, '1263818059': 1, '3952631627': 1, '3141766219': 1 },
  '3952631627': { '3951782987': 1, '2068154187': 1, '1269267787': 1 },
  '1269267787': { '3952631627': 1, '3141766219': 1 },
  '3141766219': { '1269267787': 1, '2068154187': 1, '195844427': 1 },
  '195844427': { '3141766219': 1, '1263818059': 1 }
}
//9

```

3.18 pav. Gretimumo matrica

Dvimatė kryptių matrica susideda iš į matricos celes sudėtų elementų, kurie parodo kokia kryptimi nuo pasirinktos viršūnės yra kita viršūnė. Tarkime viršūnė „2068019531“ turi viršūnę virš savęs, po apačia ir dešinėje.

```

var mapGridMatrix =
[
  [
    ['192251467', '1263818059', '195844427'],
    ['2068019531', '2068154187', '3141766219'],
    ['3951782987', '3952631627', '1269267787']
  ]
];

```

3.19 pav. Dvimatė kryptių matrica

3.9. Sistemos testavimas

3.9.1. Testavimo įrankiai ir aplinka

Centrinė valdymo sistema buvo testuojama naudojantis Mocha testavimo karkasu, kuris skirtas JavaScript kalbai, NodeJs vykdymo aplinkoje.

Roboto mechaninių elementų valdymo sistema testuota rankiniu būdu kadangi ji valdo mechaninius elementus, jos veikimo tikslumas buvo vertinamas stebint atliekamus veiksmus. Mechaniniam judėjimo tikslumui įvertinti naudojama tiesi linija ir šone nutiesta liniuotė ar kitas tikslus matavimo įtaisas. Robotui bus liepiama pajudėti tam tikrą atstumą tiesia linija ir stebima kokia yra judėjimo paklaida realiomis sąlygomis.

Priėmimo testavimas bus atliekamas realioje darbinėje aplinkoje stebint visos sistemos darbą.

3.9.2. Testavimo procedūra

3.9.2.1. Vienetų testavimas

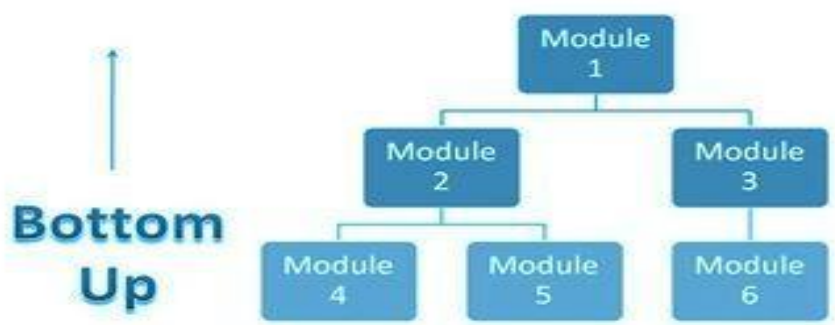
Testavimą pradėdame ne nuo visos sistemos, bet nuo atskirų komponentų testavimo (vadinamo vienetų testavimu). Vienetų testavimo etape bus testuojami atskiri roboto valdymo sistemos moduliai ir funkcijos. Taip pat bus taikomas baltos dėžės testavimo metodas, kai testuotojas žino kokį programinį kodą testuoja. Pirmiausiai bus išsiaiškinta, kokius duomenis reikia paduoti į atitinkamą programos modulį. Taip bus tikrinamos lokalių duomenų struktūros, ribinės sąlygos, skirtingi programos vykdymo keliai. Šiame testavime iš anksto laukiami rezultatai arba leistinos ir

draudžiamos išėjimo duomenų reikšmės. Tada šis testavimo procesas bus automatizuotas pasitelkiant tam skirtus įrankius ir stebimi testavimų rezultatai. Šio testavimo metu surastos klaidos turi būti ištaisomos nedelsiant.

3.9.2.2. Integravimo testavimas

Toliau komponentai palaipsniui integruojami į sistemą. Komponentų sujungimo testavimas vadinamas integracijos testavimu. Šiame etape testuojami atskiri sistemos moduliai ir sąveika tarp jų.

Atliktas testavimas gali būti laikomas testavimu iš apačios į viršų. Nors pirmiausia buvo surinktas roboto prototipas kaip esminė dalis, tačiau programinė įranga programuojama nuo mažiausio lygmens funkcijų. Tas lygmuo turi būti pilnai ištestuotas, kad būtų galima testuoti sekantį lygmenį ir pridėti aukštesnio lygio funkcionalumą į sistemą. Kadangi integracinis testavimas apima daugiau nei vien funkcijas, testavime naudojami imituojami objektai ir funkcijos, kurių konkrečiuose testuose nereikia, tačiau jie yra reikalingi sėkmingam modulio funkcijų vykdymui.



3.20 pav. Integracinio testavimo iš apačios į viršų schema

3.9.2.3. Priėmimo testavimas

Priėmimo testavimo pagrindinis uždavinys patikrinti, kaip programa atitinka programos specifikaciją. Priėmimo testavime tikrinama jau pilna programa (su visais savo moduliais ir aparatine įranga). Kadangi tikrinamas atitikimas specifikacijai, tai orientuojamasi į programos funkcionalumą.

Šiuo atveju bus vertinamas sukurtos sistemos funkcionalumas iš vartotojo perspektyvos, tikrinama ar pasiekti reikalingi funkcionalumai, kad sistema galėtų dirbti realiomis sąlygomis.

Pripažinimo testavimo metu labiau pabrėžiamas sistemos naudojamumo testavimas negu kodo prisilaikymas specifikacijos. Pripažinimo testavimas yra pagrindinis kriterijus, pagal kurį sprendžiama priimti ar atmesti sistemą.

Priėmimo testavimo metu atliekamos tokios užduotys:

- Skaičiuojamas maršrutas
- Tikrinama ar nuvažiuojama į reikiamą žymės vietą ir kaip juda robotas
- Roboto pasisukimai sankryžose
- Testuojamas kliūtis jutiklio veikimas
- Atliekama prekės paėmimo procedūra
- Vykdomas užduoties stabdymas

- Atliekama grįžimo į namų zoną procedūra
- Prekių lentynų koordinacių apskaičiavimo tikslumas

3.9.3. Testavimo rezultatai ir išvados

3.9.3.1. Vienetų testai

Įvykdžius automatinius testus, buvo gauti teigiami rezultatai – visi testai sėkmingai įvykdyti. Verta paminėti, kad šie testai buvo vykdomi nuolat, ypač kai programiniame kode atliekami pakeitimai, taip įsitikinant jog nebuvo padaryta klaidų.

```
Hex to number converter
✓ should convert hex 7B43794B to decimal number and return 2068019531
✓ should return null because of incorrect passed hex number "dsakjflafma"
✓ should return null because of passed null as parameter
✓ should return null because of passed not string value

RackMap module testing
Get positions in Matrix method
✓ finding 2068154187 and 2068019531 - should return { currentRow: 1, currentCol: 1, targetRow: 1, targetCol: 0 };
✓ finding null and null - should return { currentRow: null, currentCol: null, targetRow: null, targetCol: null };
Calculate next move direction method
✓ should return W , because target tag is in west from current
✓ should return S , because target tag is in west from current
✓ should return E , because target tag is in west from current
✓ should return N , because target tag is in west from current
checkRotation - Robot left 180degrees rotation method checker
✓ should return 5, because turning from 2068154187(EAST) to WEST so there is another path in left side
✓ should return 4, because turning from 1263818059(EAST) to WEST there is END of MAP in left side
✓ should return 4, because turning from 1269267787(EAST) to WEST there is no path in between
✓ should return 5, because turning from 195844427(NORTH) to SOUTH so there is another path in left side
✓ should return 4, because turning from 2068019531(NORTH) to SOUTH there is END of MAP in left side
✓ should return 5, because turning from 2068019531(WEST) to EAST so there is another path in left side
✓ should return 4, because turning from 3951782987(WEST) to EAST there is END of MAP in left side
✓ should return null, because tags not exist

Rack cells calculator
✓ should be number 15 thats how many cells is in array
✓ should return the exact same array of cells as expected result
✓ should return null because parameter is null
✓ should return null because parameter is not type of Array
✓ should return null because array is empty

23 passing (114ms)
```

3.21 pav. Vienetų testavimo rezultatai

3.9.3.2. Priėmimo testavimas

Šiame skyriuje aptariami priėmimo rezultatai. Atliktos užduotys ir rezultatai aprašomi žemiau esančioje lentelėje.

21 lentelė. Priėmimo testavimo rezultatas

Nr.	Užduotis	Rezultatas	Komentaras
1	Skaičiuojamas maršrutas	Maršrutas apskaičiuotas tinkamai	Maršruto skaičiavimas atliktas teisingai, surastas trumpiausias galima kelias pagal sudarytą žemėlapi
2	Patikrinti ar nuvažiuojama į reikiamą žymės vietą ir kaip juda robotas	Robotas sėkmingai nuvažiuoja į reikiama vietą ir priimtina tiesiai seka liniją	Testavimų metu pastebėta, kad 4 ratų važiuoklė dirba stabiliau nei dviejų ratų.
3	Roboto pasisukimai sankryžose	Robotas sėkmingai atlieka pasisukimus sankryžose	Pastebėta, kad robotas tiksliau atlieka pasisukimus , kai važiuoklei naudojami 4 ratai, o ne 2.

4	Kliūtis aptikimas priekyje	Kliūtis aptikimas veikia	Robotas sustoja , jei priekyje arčiau nei 20cm atsiranda kliūtis
5	Užduoties stabdymo procedūra	Procedūra veikia	Stabdymas įvykdomas sėkmingai, misija nutraukiama
6	Grįžimas į namų zoną	Procedūra veikia	Robotas sėkmingai suranda kelią iki namų vietos ir grįžta
7	Roboto judėjimas tam tikrą centimetrų kiekį, tikslumo vertinimas	<u>Judėjimas veikia su apie 5-7cm paklaida važiuojant vieną metrą</u>	Robotas sugeba pajudėti tam tikrą nustatytą atstumą, sėkmingai paskaičiuojamas reikalingas pasisukimų skaičius. Tačiau yra paklaida , kurią tikimasi bus galima spręsti programiškai arba su kokybiškesne įranga.
8	Prekių paėmimo įrenginio valdymo testavimas	Įrenginys veikia ir vykdo komandas	Įrenginį pavyksta valdyti siunčiant jam komandas iš valdymo sistemos. Vienintelis trūkumas , kad šiuo metu paėmimo įrenginio negali vežti sukurtas roboto prototipas
9	Sistemos nustatymų keitimas	Sistemos nustatymų keitimas veikia	Nustatymus galima keisti ir peržiūrėti , jie išsaugomi sistemos ilgalaikėje atmintyje
10	Prekių lentynų koordinatų apskaičiavimas	<u>Koordinatų apskaičiavimas veikia , tačiau gaunama apie 1cm paklaida</u>	Duomenų apdorojimo algoritmas veikia, tačiau paklaida yra per didelė. Ją spręsimė tobulinant duomenų apdorojimo algoritmą.

3.9.3.3. Testavimo išvados

Vykdamas programinės ir techninės įrangos testavimą buvo laikomasi apibrėžtų metodikų. Projekto vykdymo metu buvo sudarytas testavimo planas ir ištestuota sistema, aptarti jos rezultatai. Vienetų testavimas padėjo surasti defektus ir jie buvo pašalinti. Integracinis testavimas taip pat parodė spragas, kurios buvo paliktos kuriant sąveiką tarp skirtingų komponentų. Pabaigoje atliktas priėmimo testavimas leido kartu su užsakovu įvertinti projekto sėkmingumą, atrasti padarytas projektavimo klaidas bei nuspręsti tolimesnius vykdymo žingsnius.

3.10. Sistemos kūrimo rezultatas

Sistemos kūrimo metu buvo sėkmingai pasiekti užsibrėžti tikslai:

- Suprojektuotas sandėlio roboto prototipas, skirtas judėti aplinkoje.
- Sukurta programinė įranga skirta valdyti roboto techninę įrangą ir planuoti roboto užduočių veiksmus.

Roboto mechaninė valdymo sistema realizuota:

- C++ programavimo kalba
- Arduino mikrokontroleris

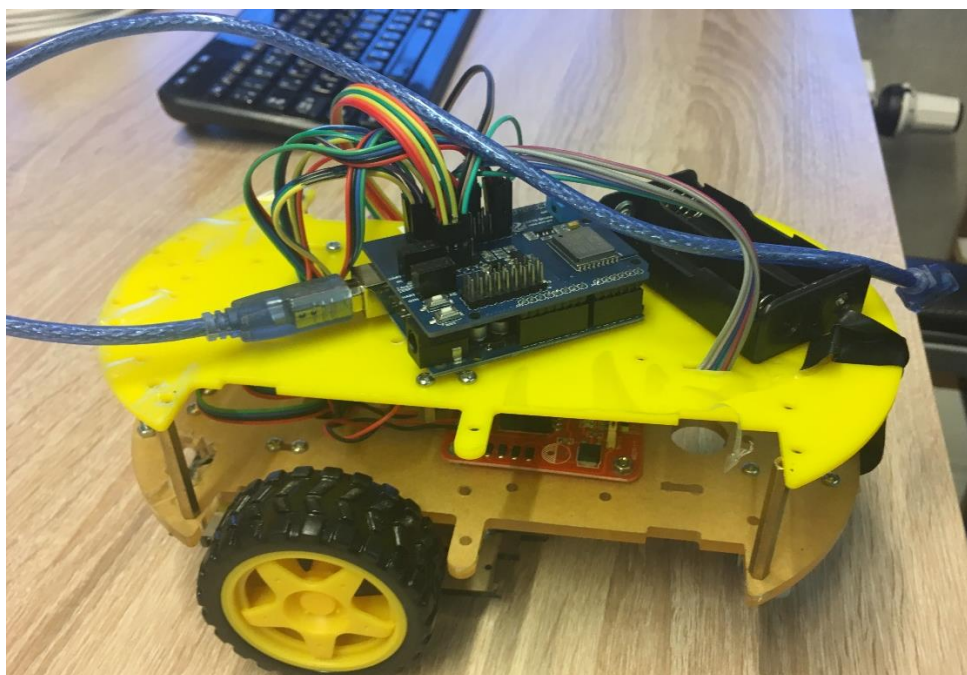
Centrinės valdymo sistemos technologijos:

- JavaScript programavimo kalba
- NodeJs vykdymo aplinka
- Raspberry Pi 3 su Raspbian OS
- Realizuota komandų sąsaja su GRBL biblioteka

Programinės įrangos kūrimo metrikos:

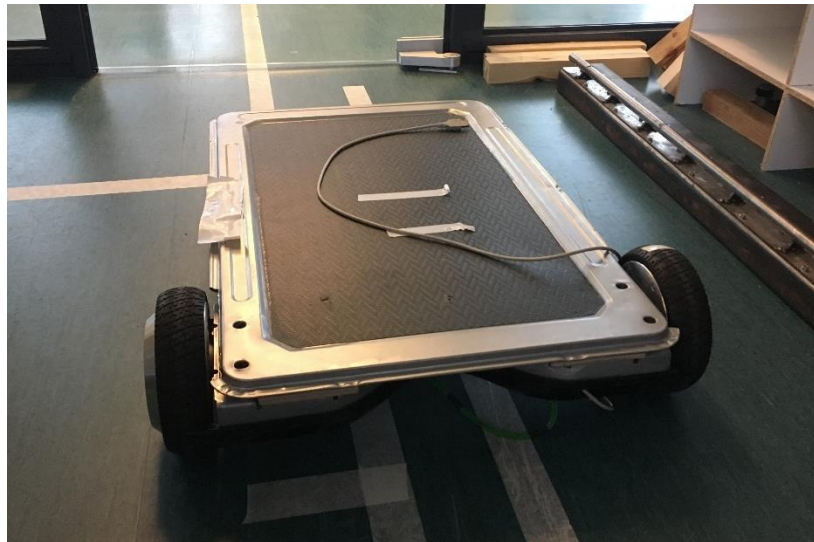
- Parašyta ~4800 eilučių kodo (neskaitant išorinių bibliotekų)
 - Centrinė valdymo sistema ~3200 eilučių
 - Roboto judėjimo sistema ~1600 eilučių
- Reikalingiems funkcionalumams pasiekti integruota 13 atvirojo kodo bibliotekų
- Sukurti 23 vienetų testai
- Rankinio testavimo metu atlikta daugiau nei 100 važiavimų vertinant tikslumą

Žemiau esančiame paveikslėlyje (žr. 3.22 pav.) pavaizduota pirmoji sutvarta roboto versija, kurios pagalba buvo testuojamas realus sistemos veikimas kuriant užduočių valdymo programinę įrangą. Komponentų sąrašas, kuris sudaro roboto techninę dalį aprašytas skyrelyje „3.10.4.4. Išdėstymo vaizdas“.



3.22 pav. Pirmoji roboto versija skirta važiavimo programinės įrangos testavimui

Toliau vykdant projektą pirmojo roboto pagrindu buvo surinktas didesnis įrenginys (žr. 3.23 pav.), kuris gali kartu ant savęs vežti ir paėmimo įrenginį (žr. 2.24 pav.). Paėmimo įrenginys bus skirtas ištraukti prekes iš lentynų.



3.23 pav. Antrosios versijos roboto prototipas



3.24 pav. Antrosios versijos roboto prototipas su uždėtu paėmimo įrenginiu

4. Tyrimo dalis

4.1. Įvadas

Vienas pagrindinių viso projekto sėkmingo veikimo reikalavimų yra sukurti įrenginio judėjimo aplinkoje tikslumas. Kadangi suprojektuotas robotas turi daug atskirų komponentų, nuo kurių priklauso bendras sistemos veikimas, būtina išsiaiškinti ir maksimaliai patobulinti įrenginio veikimą. Šio skyrelio tikslas yra ištirti svarbių roboto valdymo sistemos aplinkos suvokimo metodų ir pačio mechaninio judėjimo tikslumą. Įvertinus pradinį tikslumą bus nustatytos probleminės vietos, kurias sistemoje reikia bandyti pagerinti. Pasiūlysime ir realizuosime galimus sprendimo variantus, kurie bus tikrinami eksperimentinėje dalyje.

4.2. Pasirinkti tyrimai

Vykdamas priėmimo ir kokybės testavimą buvo pastebėtos kelios egzistuojančios problemos. Pirma, robotas nesugeba tiksliai sustoti pavažiavęs tam tikrą atstumą. Sustojimo paklaida siekia net 5-7cm važiuojant vieno metro atstumą. Sekanti problema tokia, kad įrenginiui nuskenavus prekių lentynas ir apdorojus gautus duomenis su parašytu algoritmu netiksliai nustatomos celės erdvės centrinės koordinatės. Tai pat papildomai norime paanalizuoti skirtingus variantus kaip būtų galima pagerinti prekių paėmimo įrenginį, kuris šiuo metu reikalauja papildomų komponentų. Taigi žemiau pateiktoje lentelėje (žr. 22 lentelė) pateikiamas pasirinktų tyrimų paskirtis ir aprašas.

22 lentelė. Pasirinkti tyrimai

Tyrimo numeris	Paskirtis
1	Ištirti dabartinį lentynų centro koordinatžių apskaičiavimo algoritmą, nustatyti problemas. Patobulinti esamą algoritmą pasiūlant naujus ir eksperimentų metu išbandyti tikslumą.
2	Ištirti galimas problemas kodėl robotas netiksliai nuvažiuoja jam paskirtą atstumą. Išbandyti kitokį atstumo judėjimo būdą bei įvertinti ar DC tipo varikliai yra tinkami.
3	Išbandyti skirtingus prekės paėmimo įrenginio variantus ir nustatyti, kuris geriausiai tinka dalykinėje srityje.

4.3. Tyrimui naudojama įranga

Roboto judėjimo tikslumui tirti naudojamas magistro darbe suprojektuotas prototipinis roboto modelis (žr. „3.10.4.4. Išdėstymo vaizdas“). Jame sumontuoti DC (nuolatinės srovės) tipo varikliai su enkoderiais ir H-Bridge tipo variklių kontroleris.

Koordinatžių skenavimui buvo naudotas VL53L0X lazerinis atstumo jutiklis ir realioje aplinkoje testuotos lentynos (žr. 4.1 pav.).



4.1 pav. Skenavimui ir testavimui naudotos lentynos

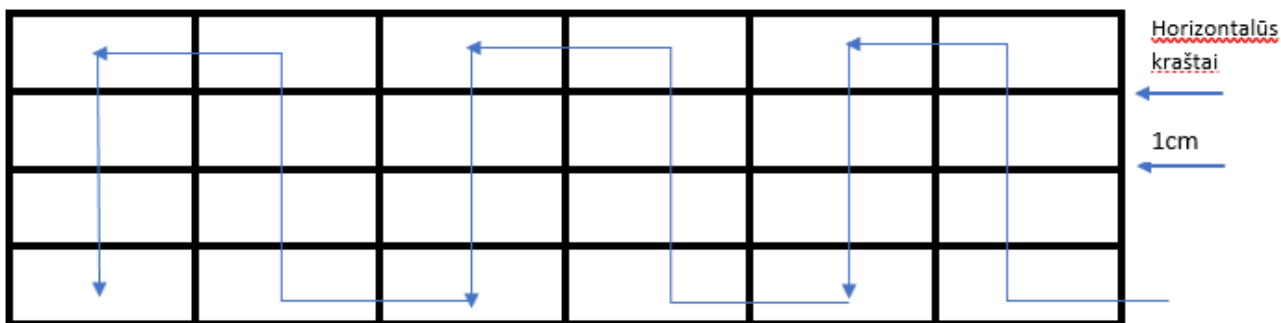
Gautų duomenų apdorojimo algoritmai išbandyti naudojant darbo metu sukurtą roboto misijų valdymo programinę įrangą, kuri buvo paleista tokioje aparatūrinėje įrangoje:

- Procesorius: Intel Core i5 1.8 Ghz
- Operatyvioji atmintis: DDR3 6Gb
- Operacinė sistema: Windows 10 64 bitų, NodeJs vykdymo aplinka

4.4. Tyrimas Nr.1

Norint išsiaiškinti galimas paklaidų priežastis skenuojant lentynas pirmiausia turime suprasti kaip visas skenavimo procesas vyksta, kokie įrenginiai čia dalyvauja bei kokie duomenys gaunami.

Taigi pirmiausia robotas atlieka vertikalų skenavimą. Įrenginys su jame esančiu lazeriniu atstumo jutikliu juda šone lentynų ir skenuoja horizontalius lentynų kraštus, kaip vaizduojama žemiau esančiame paveikslėlyje (žr. 4.2 pav.). Atstumo jutiklis geba užfiksuoti kuriuo metu robotas pravažiuoja pro lentynų kraštus, kurie pavaizduoti juodomis linijomis. Kiekvieno krašto plotis yra 1cm. Aptikus lentynos horizontalų kraštą, programinė įrangą užfiksuoja krašto koordinatas.



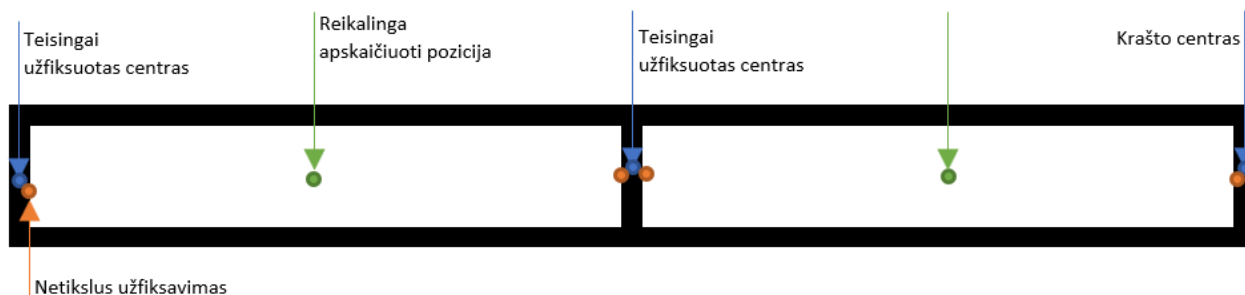
4.2 pav. Vertikalaus skenavimo schema

Atlikus pirmąjį skenavimą toliau tokiu pačiu principu vykdomas ir vertikaliųjų kraštų skenavimas (žr. 4.3 pav.). Įrenginys vėl juda per visas lentynas, tačiau dabar fiksuojami vertikalūs kraštai, kurių plotis taip pat yra 1cm. Kai lazerinis jutiklis užfiksuoja lentynos kraštą, nustatomos koordinatės, kuriose buvo aptiktas kraštas.



4.3 pav. Horizontalaus skenavimo schema

Atrodo jog skenavimas tuo ir turėtų baigtis, tačiau visa problema yra tame, kad kol jutiklis pravažiuoja 1cm kraštą, jis atsiunčia ne vieną koordinatę, o kelias. Kaip vaizduojama žemiau esančiame paveikslėlyje (žr. 4.4 pav.), mums reikalinga, kad jutiklis užfiksuoję tik patį krašto centrą, tai yra mėlynus taškelius, nes pagal juos galima tiksliai apskaičiuoti kelės centrą, kuris yra žalias taškelis. Tačiau jutiklis užfiksuoja daugiau nei vieną koordinatę (oranžiniai taškeliai) ir taip mums nėra žinoma kur yra tikrasis centras.



4.4 pav. Kelės kraštų skenavimo problema

Atliekant vienos kraštinės vertikalų skenavimą, gaunami tokie rezultatai (žr. 4.5 pav.). Kaip matome lazeriniam jutikliui judant per vieną kraštinę gaunami net 23 signalai, kad buvo užfiksuota kraštinė. Šis skaičius gali nežymiai kisti. Šiuo atveju aktualus yra Y stulpelis, nes įrenginys juda į viršų todėl kinta tik Y koordinatė. Kaip matome iš visų 23 signalų yra trys skirtingos Y reikšmės. Būtent čia ir yra problema, kuri buvo spęsta projekto metu norint nustatyti kuri iš šių trijų skirtingų koordinatė yra tikrųjų yra lentynos krašto centras.

X : Y	X : Y	X : Y
299.962: <u>54.955</u> :V	299.962: <u>62.012</u> :V	299.962: <u>68.919</u> :V
299.962: <u>54.955</u> :V	299.962: <u>62.012</u> :V	299.962: <u>68.919</u> :V
299.962: <u>54.955</u> :V	299.962: <u>62.012</u> :V	299.962: <u>68.919</u> :V
299.962: <u>54.955</u> :V	299.962: <u>62.012</u> :V	299.962: <u>68.919</u> :V
299.962: <u>54.955</u> :V	299.962: <u>62.012</u> :V	299.962: <u>68.919</u> :V
299.962: <u>54.955</u> :V	299.962: <u>62.012</u> :V	299.962: <u>68.919</u> :V
	299.962: <u>62.012</u> :V	299.962: <u>68.919</u> :V
	299.962: <u>62.012</u> :V	299.962: <u>68.919</u> :V
	299.962: <u>62.012</u> :V	299.962: <u>68.919</u> :V

4.5 pav. Skenavimo metu gautų koordinatė pavyzdys

4.4.1. Aritmetinio vidurkio metodas

Bandant iš gautų duomenų nustatyti, kuri koordinatė iš tikrųjų yra tiksli, pirmiausia buvo realizuota aritmetinio vidurkio skaičiavimo funkcija.

Šis metodas veikia labai paprastai, ima visas sienelės koordinates, kurios pavaizduotos buvo anksčiau ir suskaičiuoja vidurkį taip tikintis, kad rezultatas ir bus pats sienelės vidurys.

Tačiau šis metodas pasirodė labai jautrus nuskaitytomis reikšmėms, pastebėtos problemos:

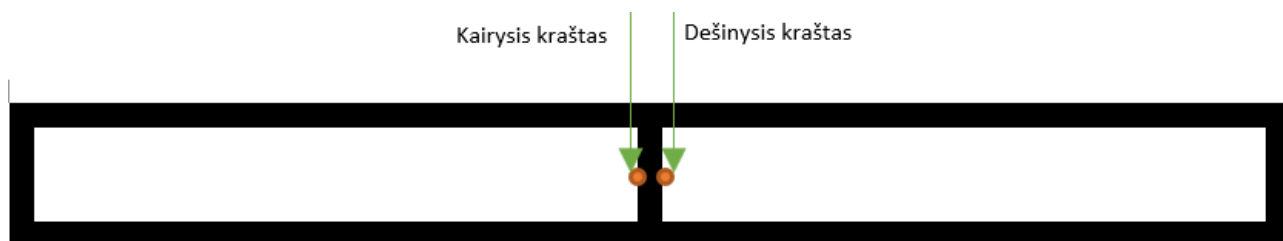
- Jei jutiklis netyčia užfiksuoja bent vieną didesnę ar mažesnę reikšmę, tarkime prie dabartinių turimų reikšmių 54.955, 62.012, ir 68.919 atsiranda viena neteisingai užfiksuota reikšmė 78, tai vidurkio reikšmė nemažai išsikraipo.
- Vidurkio rezultatui didelę reikšmę turi ir tai, kiek visoje imtyje buvo pasikartojančių konkrečių elementų. Jei konkrečiame skenavime kažkuri koordinatė buvo užfiksuota labai daug kartų, kita mažai.
- Negalėjome teigti, kad vidurys turėtų būti ten, kur koordinatės pasikartojimų buvo daugiausia, nes kai kuriais atvejais jutiklis užfiksuodavo skirtingas koordinates po tiek pat kartų.

Bandant išspręsti pastebėtą problemą sugalvota realizuoti du papildomus algoritmus, kurių pagalba siekiama pagerinti tikslumą.

- Didžiausios-mažiausios reikšmės metodas
- Dažniausio pasikartojimo metodas

4.4.2. Didžiausios-mažiausios reikšmės metodas

Vienas iš sugalvotų ir realizuotų metodų yra didžiausios ir mažiausios reikšmių metodas. Šio metodo mintis yra tokia, kad koordinacių imtyje esanti mažiausia reikšmė turėtų būti vienas nuskenuotos sienelės kraštas, o didžiausia reikšmė turėtų būti kitas kraštas. Taip gautume abiejų sienelės tolimiausių kraštų koordinates, o tada sudėjus abidvi reikšmes bei padalinus iš dviejų gautume vidurinį tašką, kurio ir ieškome.



4.6 pav. Metodo norimų gauti reikšmių vizualizacija

Jei pasižiūrėtume pateiktą pavyzdį (žr. 4.5 pav.), tai čia mažiausia reikšmė būtų 54.955 , o didžiausia 68.919. Taigi krašto centras būtų koordinatėje 61.937. Išbandysime šį metodą eksperimentinėje dalyje ir įvertinsime tikslumą.

4.4.3. Dažniausio pasikartojimo metodas

Dar vienas iš realizuotų ir bandytų variantų yra dažniausio pasikartojimo metodas. Šio metodo idėja yra iš nuskenuotų duomenų išrinkti dominuojančių koordinacių porą ir tada suskaičiuoti vidutinę reikšmę. Tokiu išskyrimu yra siekiama kuo tiksliau nustatyti, kurios iš tikrųjų yra tikslios koordinatės ir eliminuoti netikslius matavimus - tai yra pašalinius koordinacių užfiksavimus.

Šios problemos pavyzdys yra pavaizduotas žemiau esančiame paveikslėlyje (žr. 4.7 pav.). Skenuojant sienelę buvo užfiksuotos net keturios skirtingos koordinatės 855.706, 863.063, 870.12 ir 877.027. Tačiau kaip matome mažiausia ir didžiausia reikšmės buvo užfiksuotos tik po vieną kartą, iš to galime įtarti, kad tai yra tik netikslus jutiklio nuskaitymas. Taigi šis realizuotas algoritmas ir spręs šią problemą. Pirmiausia bus atrenkami du koordinacių rinkiniai, kurių pasikartojimo skaičius didžiausias. Tai mums turėtų leisti paimti sienelės vietą, kuri tikrai yra arčiausiai centro. Paskutiniame žingsnyje iš dviejų koordinacių bus paskaičiuojama vidutinė reikšmė.

X : Y	X : Y	X : Y	X : Y
299.962: <u>855.706</u> :V	299.962: <u>863.063</u> :V	299.962: <u>870.12</u> :V	299.962: <u>877.027</u> :V
	299.962: <u>863.063</u> :V	299.962: <u>870.12</u> :V	
	299.962: <u>863.063</u> :V	299.962: <u>870.12</u> :V	
	299.962: <u>863.063</u> :V	299.962: <u>870.12</u> :V	

4.7 pav. Pašalinių koordinacių pavyzdys

4.4.4. Apibendrinimas

Taigi skenavimo tikslumo tyrimo metu pasiaiškinome galimas netikslaus skenavimo problemas ir realizavome du papildomus algoritmus. Bendra visų algoritmų veiklos diagrama pavaizduota skyrelyje „3.10.4.2 Veiklos diagramos“. Algoritmų veikimas ir rezultatai bus aprašytas eksperimentinėje dalyje.

4.5. Tyrimas Nr.2

Vienas svarbiausių viso projekto sėkmės faktorių yra tikslus roboto judėjimas aplinkoje. Viena iš reikalingų roboto funkcinių reikalavimų yra sugebėti pavažiuoti tam tikrą atstumą ir tiksliai sustoti. Šiuo metu surinktas prototipas naudoja DC (nuolatinės srovės) tipo variklius su enkoderiais. Būtent su enkoderių pagalba ir iš anksto apskaičiuojant reikalingą variklių apsisukimų skaičių prototipinėje versijoje bandoma nuvažiuoti tam tikrą atstumą.

Norint sužinoti kokį atstumą robotas nuvažiavo, turime išsiaiškinti kaip suprasti turimo variklio enkoderio tikų (angl. ticks) duomenis. Šiuo atveju variklio ir enkoderio duomenys yra tokie:

- Variklio perdavimo skaičius (angl. gear ratio) – 120:1
- Enkoderio skaičius (angl. encoder ratio) – 16:1
- Rato skersmuo - 60mm

Turėdami šiuos duomenis galime padaryti tokias išvadas:

- 1 pilnas rato apsisukimas = 120 motoro apsisukimų
- 1 motoro apsisukimas = 16 enkoderio tikų
- 1 pilnas rato apsisukimas = 120 motoro apsisukimų * 16 enkoderio tikų = 1920 tikų

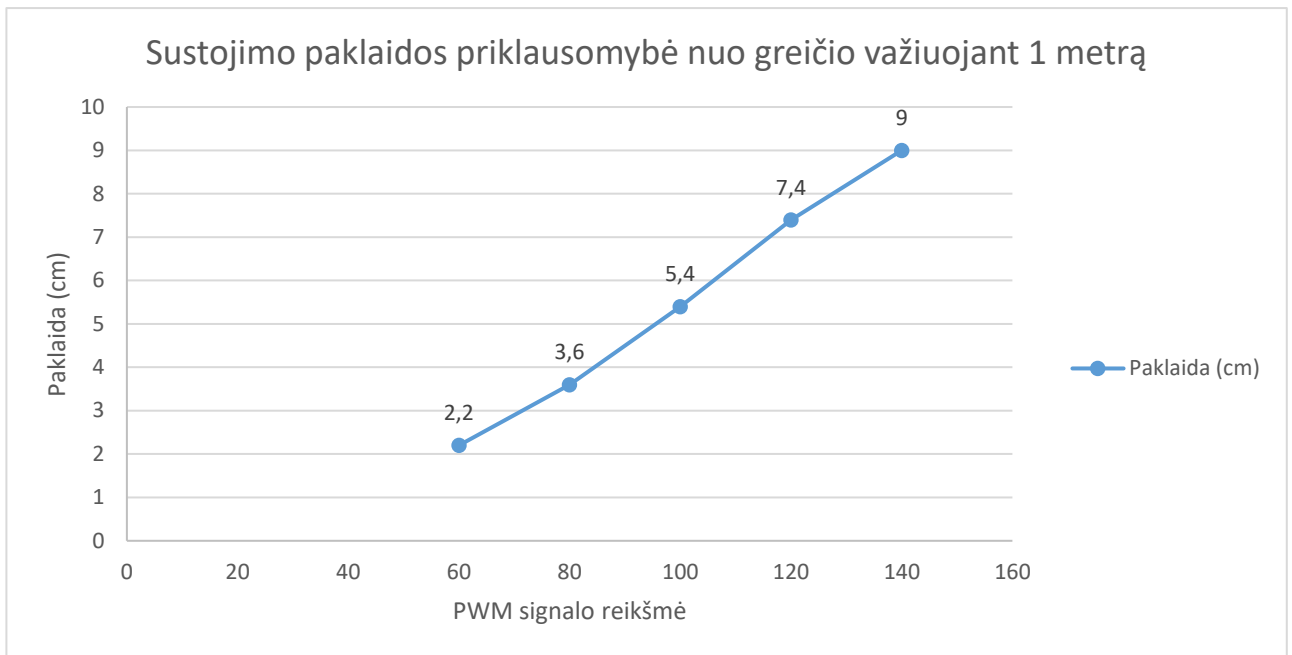
Iš šių paskaičiavimų nustatyta, kad per vieną rato apsisukimą turimas enkoderis į valdymo programinę įrangą nusiųs 1920 tikų. Toliau galima suskaičiuoti kiek vienas tikas atitinka realaus nuvažiuoto atstumo erdvėje:

- 1 rato apsisukimas = π * rato skersmuo = 3.14 * 60mm = 188,4mm
- 1920 tikų = 188,4mm
- 1 tikas = 0.098125mm

Taigi apskaičiavę gauname, kad vienas enkoderio tikas atitinka 0.098125mm nuvažiuoto roboto kelio. Turint tokius duomenis galima apskaičiuoti kiek ilgai variklių valdiklis turi laikyti įjungtus motorus ir kada išjungti.

4.5.1. Judėjimas be stabdymo

Anksčiau apskaičiuoti judėjimo duomenys variklių valdiklyje leido realizuoti važiavimo funkciją, kuri turėtų pajudinti robotą tam tikrą atstumą ir tada sustoti. Tačiau atliekant kokybės analizę pastebėtas, kad robotui važiuojant nustatytą atstumą atsirado gana didelė ir mums nepriimtina paklaida (žr. 4.8 pav.).



4.8 pav. Sustojimo paklaida priklausomai nuo greičio

Kaip matome iš pateikto grafiko, didėjant roboto greičiui didėja ir paklaida, tai yra robotas pravažiuoja sustojimo tašką. Mažiausia PWM signalo reikšmė yra 60, nes tai mažiausias dydis, su kurio robotas dar geba pajudėti iš vietos. Standartinis greitis, kuriuo norima, kad robotas judėtų yra 100. Taigi važiuojant standartiniu greičiu paklaida yra apie 5,4cm.

Išnagrinėjus grafiką ir judėjimą, padaryta prielaida, kad sustojimo taške nepakanka tik išjungti variklius ir tikėtis jog įrenginys tiksliai sustos. Neįvertinome, kad kuo greičiau juda įrenginys, tuo didesnę inerciją jis turi. Kadangi šiuo atveju mechaninių stabdžių įrenginys neturi, bandėme išsiaiškinti ar įmanoma sustoti reikiamoje vietoje roboto stabdymą kontroliuojant vien iš programinės pusės. Išbandyti du papildomi įrenginio stabdymo metodai:

- Palaipsninis greičio mažinimas artėjant prie tikslo
- Stabdymas varikliu į priešingą judėjimui pusę

4.5.2. Palaipsninis greičio mažinimas

Šio stabdymo metodo esmė, pasistengti kuo labiau sulėtinti robotą prieš sustojimo tašką. Kadangi bandymų metu išsiaiškinome, jog roboto stabdymo kelias priklauso nuo greičio ir inercijos reikia pabandyti po truputį mažinti roboto judėjimo greitį iki minimalaus jam artėjant prie tikslo.

Visa tai galime įgyvendinti suprogramuojant tam tikrus tarpinius atstumus, kuriuose robotas privalo neviršyti nustatyto greičio. Bandymų metu nustatėme tokias ribas: pirmus 80% robotas juda standartiniu greičiu, likus 20% iki tikslo sulėtėja iki 80pwm ir likus 10% sulėtėja iki 60pwm, mažiausios greičio ribos, kuri dar paveža įrenginį.



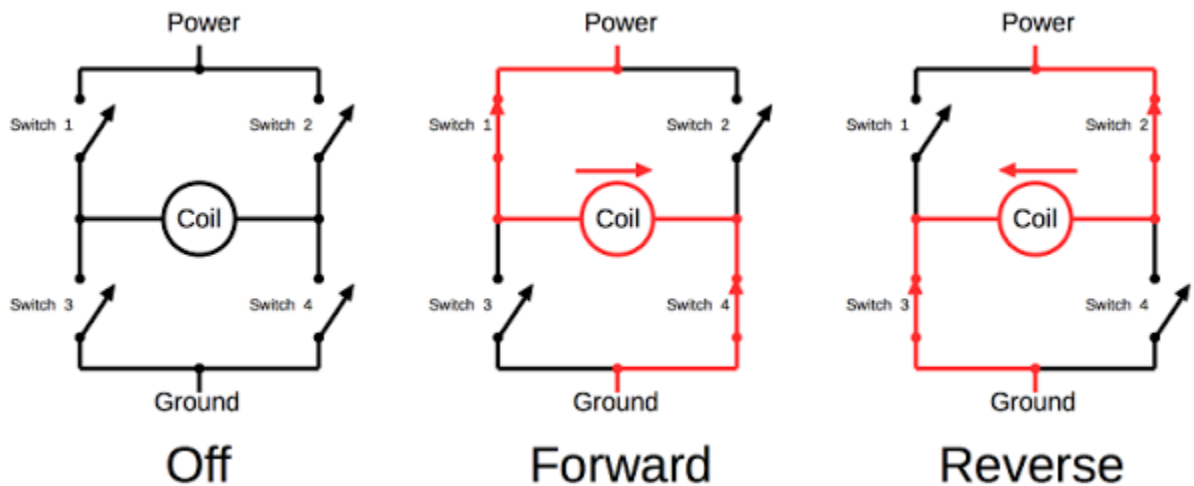
4.9 pav. Roboto greičio mažinimo procesas

4.5.3. Lėtėjimas ir variklio blokavimas

Dar vienas būdas, kurį išbandysime yra palaipsninio stabdymo metodas papildytas staigiu variklio stabdymu. Toks efektas pasiekiamas ne vien tik lėtėjant ir išjungiant įtampą varikliams, tačiau paduodant įtampą priešinga kryptimi ir taip priverčiant variklį labai trumpą laiko tarpą sukstis į priešingą pusę blokuojant judėjimą į priekį. Toks veiksmas blokuoja ratus ir neleidžia jiems iš inercijos judėti tolyn. Norint pasiekti tokį efektą vykdoma tokia veiksmų seka:

- Robotas juda pastoviu greičiu
- Robotas lėtėja
- Pasiekus pabaigos tašką išjungiami varikliai
- Iškart trumpam laiko tarpui nustatomas įtampos tekėjimas priešinga kryptimi, kas privers ratus sukstis atgal

Tai galima pasiekti naudojant H-Bridge tipo variklių kontrolerį (žr. 4.10 pav.). Keičiant įtampas skirtinguose valdiklio vartuose (angl. *gates*), keičiasi ir ratų sukimosi kryptis.



4.10 pav. H-Bridge tipo variklių valdiklio krypčių schema

Šis būdas nėra pats geriausias, nes per didelė elektros srovė stabdymo metu gali sugadinti elektronikos komponentus. Reikia atidžiai įvertinti kaip ilgai galima būti padidinus srovę. Tačiau šiuo atveju jis tiktų, nes srovės nėra didelės ir bandomas pats principo veiksmingumas.

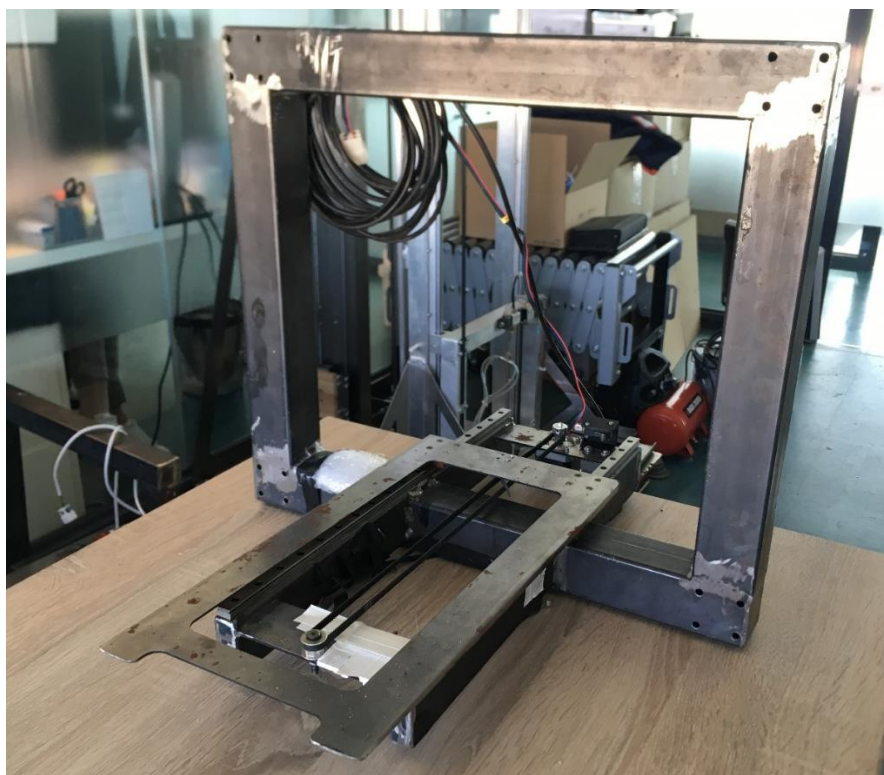
4.6. Tyrimas Nr.3

Paskutiniojo tyrimo tikslas yra išbandyti turimą prekėms paimti skirtą komponentą, išvelgti jo privalumus bei trūkumus. Tuomet pabandydysime aptarti kitus galimus prekių paėmimo būdus, kurie tiktų dalykinėje srityje ir galėtų būti pritaikomi prie esamos įrangos.

Kad šis judantis robotas galėtų paimti prekes iš lentynų, jis turi turėti tam tikrą mechanizmą, kuris gebėtų ištraukti ar kitaip paimti lentynose padėta prekę. Pirmasis įrenginys, kuris buvo pagamintas projekto metu buvo į šoną išvažiuojantis „liežuvėlis“, jis aprašomas žemiau esančiame skyrelyje „4.6.1 Išsiskleidžiantis paėmimas“ .

4.6.1. Išsiskleidžiantis paėmimas

Paėmimo įrenginys sudarytas iš judančios platformos, vadinamojo „liežuvėlio“. Žingsninio variklio pagalba judanti platforma gali išvažiuoti į priekį, pakelti ir įtraukti atgal reikalingą objektą.



4.11 pav. Išsiskleidžiantis paėmimo įrenginys

Kad šis įrenginys sugebėtų paimti prekę, jis būtinai turi būti komplektuojamas su papildomu padėkliuku (žr. 4.12 pav.) ant kurio viršaus uždedama prekės. Tokiu būdu paėmimo įrenginio platforma gali įvažiuoti ir pakelti padėkliuką su ant jos stovinčia preke bei ją ištraukti.



4.12 pav. Išsiskleidžiančio paėmimo padėkliukas

Testuojant įrenginio veikimą buvo pastebėti tam tikri trūkumai:

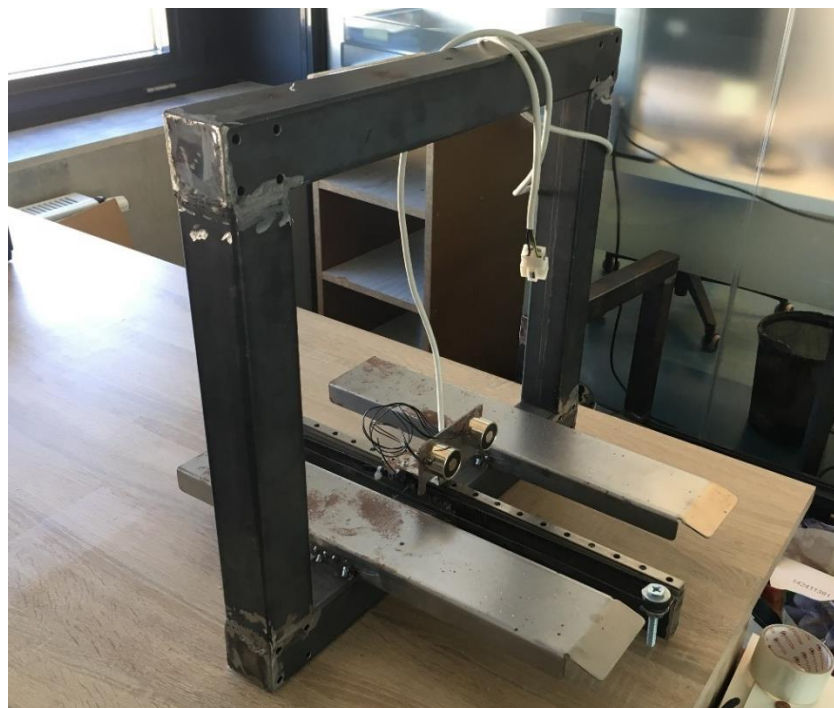
- Reikia tiksliai žinoti, kurioje pozicijoje yra padėkliuko centras, nes kitaip paėmimo įrenginys nepataiko į jį įvažiuoti. Taigi reikalingas didelis tikslumas.
- Kiekviena prekė turi būti padėta ant padėkliuko, tai apsunkina sistemos pritaikymą, mažina lankstumą ir didina eksploatacijos kainą.

Siekiant išspręsti pirmojo įrenginio problemas ir patobulinti paėmimo procesą išbandysime du kitokius paėmimo variantus:

- Magnetinis paėmimas
- Vakuuminis paėmimas

4.6.2. Magnetinis paėmimas

Magnetinio paėmimo įrenginys taip pat turi išvažiuojantį mechanizmą, tačiau ant jo galo yra pritvirtintas magnetas, kuris pritraukia metalinį paviršių prie savęs ir tada galima jį ištraukti iš lentynos.



4.13 pav. Magnetinis paėmimo įrenginys

Kadangi prekės nebūna metalinės ir paprastai šis įrenginys prekės negalės pritraukti, jam taip pat yra būtina naudoti papildomą priedą. Šiam paėmimo būdai pritaikytas priedas yra metalinis padėkliukas (žr. 4.14 pav.), kurio priekyje yra speciali metalinė plokštelė. Magnetis prisilietęs prie šios plokštelės pritraukia ją ir taip gali kartu ištraukti ant padėkliuko padėtą prekę.



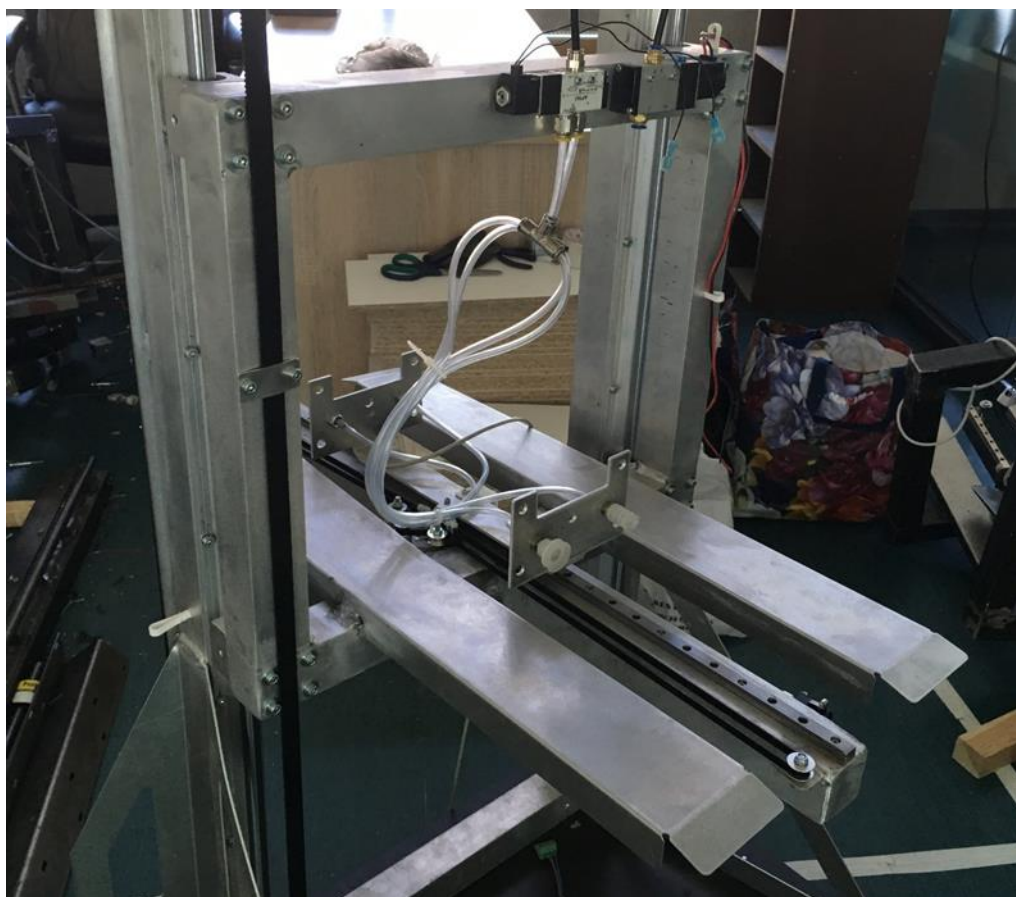
4.14 pav. Magnetinio paėmimo įrenginio padėkliukas prekei

Nors šis paėmimo metodas taip pat reikalauja papildomos įrangos (padėkliuko) sėkmingam naudojimui, tačiau privalumas lyginant su pirmuoju turėtų būti tas, kad šis padėkliukas su priekyje

esančia didele plokšte sumažina poreikį labai tiksliai privažiuoti prie prekės, nes paėmimo plotas yra didesnis.

4.6.3. Vakuuminis paėmimas

Dar vienas įrenginys, kurio tinkamumą vertinsime eksperimentinėje dalyje yra vakuomo principu paremtas prekių paėmimas. Šis paėmimo įrenginys (žr. 4.15 pav.) taip pat kaip ir kiti turi išvažiuojančią platformą, tačiau šiuo atveju pagrindinė dalis yra siurbtukai, kuriuose išsiurbiant orą sukuriamas vakuumas, skirtas pritraukti objektą.



4.15 pav. Vakuuminis paėmimo įrenginys

Šio įrenginio privalumas turėtų būti tas, kad jame nereikalingi papildomi priedai (padėkliukai) ant kurių reikia dėti prekę, taigi sumažėja sąnaudos bei lengviau pritaikyti dalykinėje srityje. Prekių barkodų niekas neuždengia. Kita vertus šis sprendimas reikalauja papildomos įrangos, kuri sugebėtų siurbti orą ir sukurti vakuumą reikalingą prekių paėmimui.

5. Eksperimentinė dalis

Eksperimento metu bus išbandomi tyrimo dalyje pasiūlyti problemų patobulinimai. Čia atlikti eksperimentai padės įvertinti ar pasiūlyti sprendimai atnešė naudos bei parodys kokia linkme toliau reikėtų vystyti projektą. Kaip ir tyrimo dalyje, čia bus atliekami trys eksperimentai – po vieną kiekvienai tyrimo daliai (žr. 22 lentelė).

5.1. Eksperimentas Nr.1

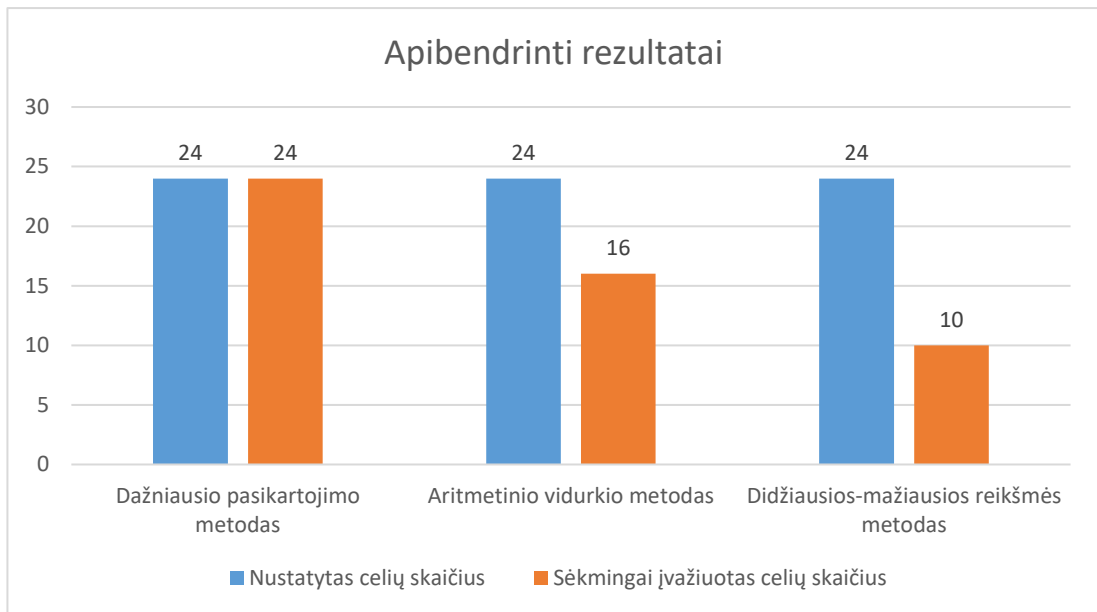
Pirmojo eksperimento metu išbandysime senojo lentynų skenavimo algoritmo ir dviejų naujų pasiūlytų metodų tikslumą vertinant nustatytų koordinačių nukrypimus nuo tikslų kiekvienos celės centro koordinačių. Tai mums leis įvertinti, kuris algoritmas geriausiai ir tiksliausiai paskaičiuoja celių centrus ir kiek celių iš viso aptinka. Informacija apie tyrimą:

- Eksperimento metu buvo atlikti 3 pilni lentynų skenavimai. Lentynos pavaizduotos skyrelyje „4.4. Tyrimas Nr.1“.
- Iš kiekvieno skenavimo vidutiniškai gauta 1010 koordinačių.
- Skenuotas lentynas sudarė 6 stulpeliai ir 4 eilės – viso 24 celės, kurias algoritmai turi aptikti ir suskaičiuoti jų centro koordinatas.
- Skenavimo greitis 5cm/s

Realizavus tyrimo dalyje aptartus metodus atliktas eksperimentinis tikslumo tyrimas. Su kiekvienu metodu nustatėme lentynoje esančių celių skaičių ir bandėme realiai į jas įvažiuoti. Apibendrinti rezultatai pateikti žemiau 23 lentelėje ir 5.1 paveikslėlyje.

23 lentelė. Eksperimento Nr.1 apibendrinti rezultatai

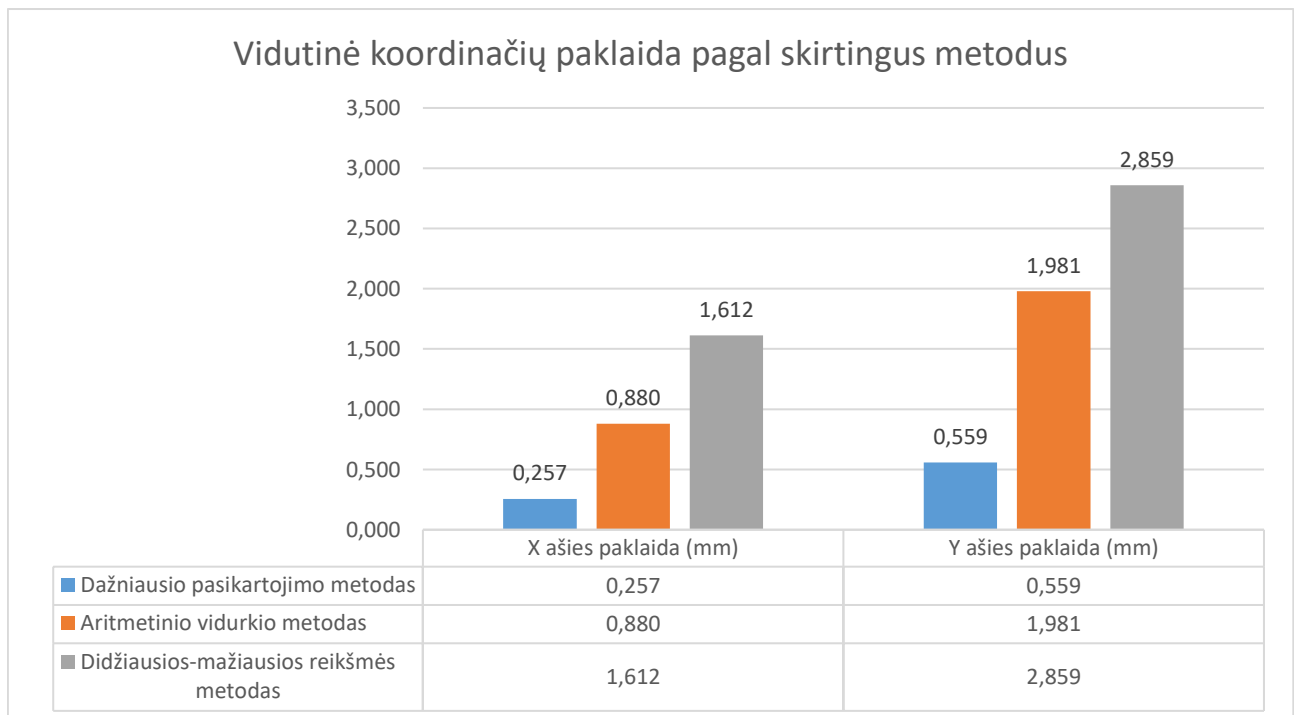
	Dažniausio pasikartojimo metodas	Aritmetinio vidurkio metodas	Didžiausios-mažiausios reikšmės metodas
Nustatytas celių skaičius	24	24	24
Sėkmingai įvažiuotas celių skaičius	24	16	10



5.1 pav. Tyrimo Nr.1 apibendrinti rezultatai

Kaip matome visi testuoti algoritmai sėkmingai suskaičiavo, kad naudotose lentynose yra 24 celės. Tačiau realiai bandant įvažiuoti į kiekvieną iš nustatytų rezultatai skyrėsi. Didžiausios-mažiausios reikšmės metodas parodė blogiausius rezultatus, nes pavyko įvažiuoti tik į 10 celių. Aritmetinio vidurkio metodas parodė geresnius rezultatus, pataikė į 16 celių. Idealius rezultatus parodė dažniausio pasikartojimo metodas, tiksliai įvažiuota į visas 24 celes.

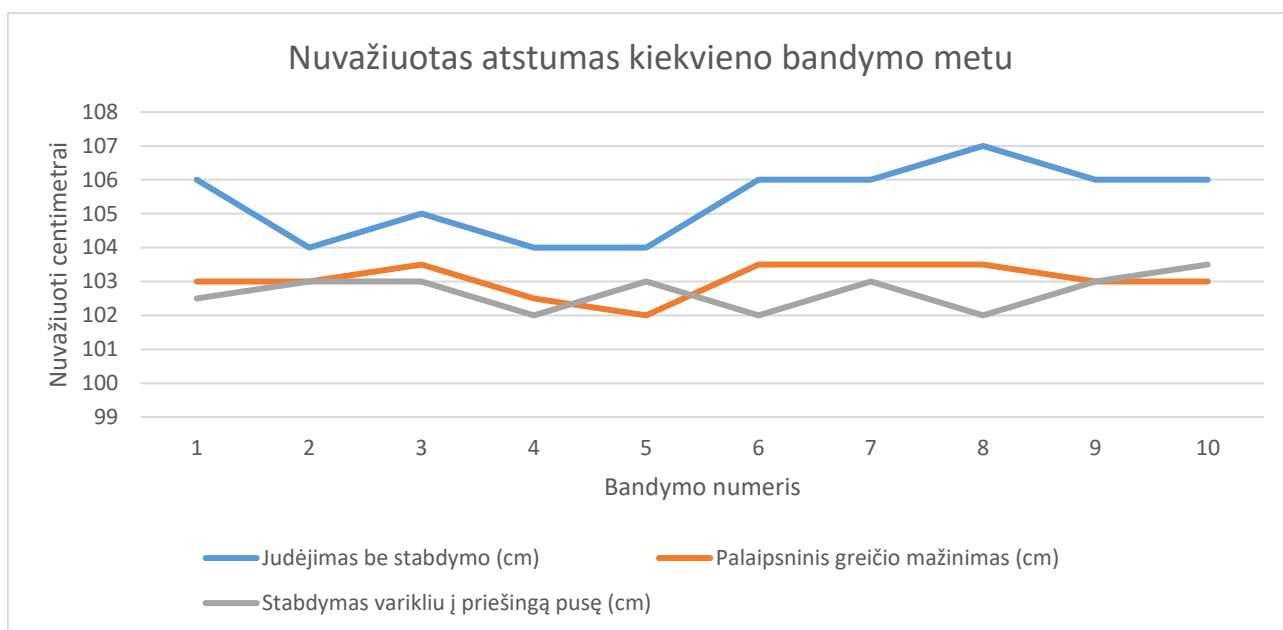
Kad suprasti kodėl skiriasi sėkmingai įvažiuotų celių skaičius, galime pažiūrėti į kiekvieno algoritmo nustatytų koordinatų vidutines paklaidas (žr. 5.2 pav.).



5.2 pav. Vidutinė koordinatų paklaida pagal skirtingus metodus

Diagramoje matome, kad lyginant su išmatuotomis celių centrų koordinatėmis mažiausias paklaidas parodė būtent dažniausio pasikartojimo metodas. Jo paklaidos vidutiniškai siekė tik 0.257mm X ašyje ir 0,559mm Y ašyje. Šiuo metodu pavyko išgauti tikslumą su mažesne nei vieno milimetro paklaida. Kituose metoduose labiausiai išryškėja didelė Y ašies paklaida, būtent ji ir buvo didžiausia problema realių testavimų metu, kai paėmimo įrenginys privažiudavo per žemai arba per aukštai, kad būtų pataikyta į tikslią prekės paėmimo vietą.

Nustatyti jog dažniausio pasikartojimo metodas yra tiksliausias ir labiausiai artimas realioms duomenims padėjo ir žemiau esanti diagrama (žr. 5.3 pav.). Šiame pavyzdyje pateikta 6 celių Y ašies koordinatinių reikšmės. Kadangi tikslų koordinatinių juosta grafike yra geltona, tai kuo arčiau yra kitų spalvų juostos, tuo rezultatai yra artimesni realioms reikšmėms. Taigi čia taip pat matome, kad beveik identiška su geltona juosta eina ir mėlyna, kuri vaizduoja dažniausio pasikartojimo metodą.



5.3 pav. Skirtingų metodų rezultatų palyginimas su realiais duomenimis

Įvertinus eksperimento rezultatus matome, kad tyrimo darbe pasiūlyti sprendimai tikslumo problemai išspręsti - suveikė. Panaudojus dažniausio pasikartojimo metodą pavyko pasiekti mažesnę nei 1mm nukrypimą nuo realių pozicijų, todėl šis metodas bus naudojamas ir tolimesnėse kūrimo stadijose.

5.2. Eksperimentas Nr.2

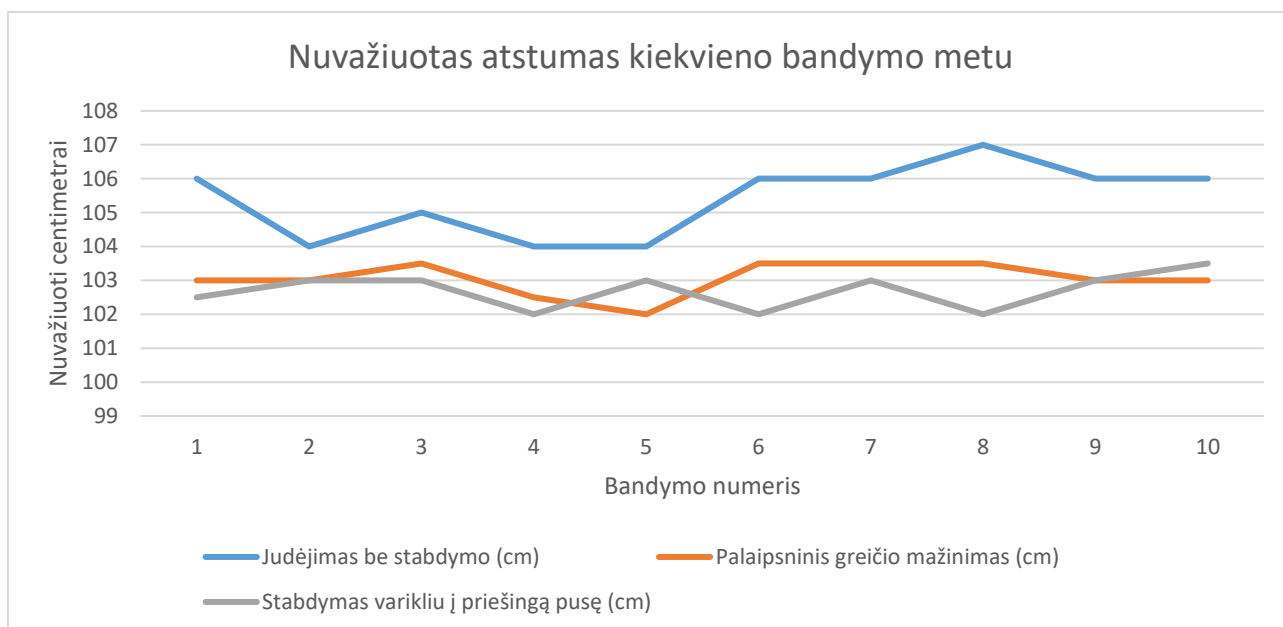
Antrojo eksperimento tikslas išbandyti naujus metodus, kurie buvo sukurti siekiant pagerinti robotų stabdymo tikslumą, nes kaip tyrimo dalyje nustatėme, robotas stipriai viršina paklaidas, kuriose turėtų sustoti. Taigi šis eksperimentas padės nustatyti ar pakeitimai ir realizuoti metodai veikia sukurtoje sistemoje.

Bandymų metu su kiekvienu stabdymo metodu buvo atlikta po 10 važiavimų. Kiekvieno važiavimo metu roboto tikslas nuvažiuoti lygiai 1m ir sustoti. Robotų judėjimo greitis testavimo metu buvo 15cm/s. Žemiau esančioje lentelėje pateikti bandymų duomenys parodo kokį atstumą kiekvieno bandymo metu nuvažiuo robotas priklausomai nuo skirtingų metodų.

24 lentelė. Eksperimento Nr.2 bandymų rezultatai

	Judėjimas be stabdymo (cm)	Palaipsninis greičio mažinimas (cm)	Lėtėjimas ir variklio blokavimas (cm)
1	106	103	102,5
2	104	103	103
3	105	103,5	103
4	104	102,5	102
5	104	102	103
6	106	103,5	102
7	106	103,5	103
8	107	103,5	102
9	106	103	103
10	106	103	103,5
Vidurkis	105,4cm	103,05cm	102,7cm

Kaip matome lentelėje ir žemiau pateiktame grafike, labiausiai netikslus stabdymo metodas yra „Judėjimas be stabdymo“, nes stabdant šiuo metodu robotas pravažiudavo daugiausiai. Bandant taikyti kitus du pasiūlytus metodus tikslumas pagerėjo. Abu metodai „Palaipsninis greičio mažinimas“ ir „Lėtėjimas ir variklio blokavimas“ parodė panašius rezultatus, tačiau pastarasis buvo šiek tiek tikslesnis. Vertinant vidutiniškai nuvažiuotą atstumą metodus „Lėtėjimas ir variklio blokavimas“ nuvažiuodavo 102,7cm, tai reiškia, kad būdavo padaroma vidutiniškai 2,7cm paklaida.



5.4 pav. Nuvažiuoto atstumo diagrama kiekvieno bandymo metu

Taigi atlikus eksperimentus išsiaiškinome, kad pavyko pagerinti stabdymo tikslumą. Geriausias rezultatas buvo pasiektas naudojant „Lėtėjimas ir variklio blokavimas“ metodą. Vis dėl to,

eksperimentų rezultatai nepasiekė tokių reikšmių kokios reikalingos, kad robotas gebėtų tiksliai atlikti savo užduotis.

Gavus neigiamus rezultatus buvo bandyta patestuoti judėjimą naudojant BLDC (bešepetėlinius nuolatinės srovės) tipo variklius, tačiau jie taip pat parodė blogus rezultatus.

5.3. Eksperimentas Nr.3

Šio eksperimento metu buvo vertinami tyrimo dalyje aprašyti paėmimo įrenginiai. Eksperimento tikslas realiai išbandyti kiekvieną įrenginį ir įvertinti, kuris geriausia atitinka dalykinės srities poreikius.

Įvertinimui buvo paruošta kokybės vertinimo rezultatų lentelė (žr. 25 lentelė.), kuri apima svarbiausius aspektus pagal kuriuos galime korektiškai įvertinti kiekvieną iš įrenginių. Vertinama bus balais nuo 1 iki 5. Labiausiai tinkamas įrenginys bus tas, kurio balų skaičius didžiausias

25 lentelė. Eksperimento Nr.3 įrenginių vertinimo kriterijų lentelė

Kriterijai	Vertinimo detalizavimas	Paaiškinimas	Vertinimas (nuo 1 iki 5)
Tikslumas	Reikia įvertinti kaip tiksliai sistema turi sugebėti privažiuoti prie prekės naudojant įrenginį.	Skirtingi įrenginiai turi skirtingą paėmimo procedūra bei paviršiaus plotą. Kuo reikalingas didesnis tikslumas, tuo sunkiau pritaikyti.	1 – reikalingas didelis tikslumas 5 – reikalingas mažas tikslumas
Kaina	Reikia įvertinti kiek kainuoja tokio įrenginio komponentai ir gamyba.	Kiekvienas įrenginys turi tam tikrus specifinius elementus, kad galėtų veikti. Įvertinti jų kainos įtaką.	1 – kaina didelė 5 – kaina maža
Pritaikomumas	Reikia įvertinti kiek pastangų reikia, kad įrenginys galėtų veikti pasikeitusioje aplinkoje.	Vertinama kiek pastangų reikia, kad įrenginys galėtų paimti kitokio tipo prekes (pasikeičia dydis, pakuotė)	1 – sunku pritaikyti 5 – lengva pritaikyti
Suderinamumas	Įvertinti ar įrenginys ir jo komponentai nesukelia trikdžių kitoms sistemoms	Vertinama ar paėmimo įrenginio specifiška neapsunkina kitų sistemų darbo	1 – konfliktai su kitais sistemos komponentais 5 – konfliktų nėra
Greitis	Reikia įvertinti kaip greitai konkretus įrenginys gali atlikti prekės paėmimo veiksmą.	Įvertinama ar tarp įrenginių yra esminiai greičių skirtumai.	1 – greitis mažas 5 – greitis didelis
Papildoma įranga	Reikia įvertinti kiek papildomos įrangos reikia sėkmingam įrenginio funkcionavimui.	Įrenginius gali būti reikalinga komplektuoti su papildomais komponentais. Kuo tokių dalykų daugiau, tuo kaina didesnė	1 – reikalinga daug įrangos 5 – įranga nereikalinga

Pagal aukščiau aprašytus kriterijus įvertinome kiekvieną iš paėmimo būdų:

- Išsiskleidžiantis paėmimas
- Magnetinis paėmimas
- Vakuuminis paėmimas

Išsiskleidžiantis paėmimas

Šis įrenginys reikalauja didžiausio tikslumo iš visų, nes padėkliuko ertmė į kurią reikia pataikyti yra tik 2cm aukščio. Kainos atžvilgiu pats paėmimo įrenginys neturi išsiskiriančių komponentų, todėl gamybos kaina standartinė. Pritaikomumas yra viena iš keblių vietų šio tipo sistemoje, nes pasikeitus prekės dydžiams taip pat turi būti pritaikomi padėkliukai, kurie gaminai iš naujo, kita vertus įrenginys gali puikiai dirbti su skirtingų medžiagų ar formų daiktai. Išsiskleidžiančio paėmimo įranga netrikdo kitų įrenginių darbo. Greičio atžvilgiu šis būdas yra našiausias ir gali judėti didžiausiu greičiu iš visų, tačiau viena didžiausių problemų yra ta, kad reikalinga papildoma įranga, tai yra padėkliukais kiekvienai prekei, tai stipriai padidina kainą, sumažina lankstumą ir pritaikomumą. Apibendrinta įverčių lentelė žemiau.

26 lentelė. Išsiskleidžiančio paėmimo vertinimo rezultatas

	Tikslumas	Kaina	Pritaikomumas	Suderinamumas	Greitis	Papildoma įranga
Įvertis	1	4	3	5	5	2

Magnetinis paėmimas

Magnetinis įrenginys reikalauja mažesnio tikslumo nei išsiskleidžiantis, nes magneto prisilietimo plotas yra didesnis. Kainos atžvilgiu šis įrenginys niekuo neišsiskiria, tik papildomus kaštus sudaro magnetinis įrenginys. Pritaikomumo aspektas labai panašus į prieš tai aptarto įrenginio. Kadangi čia taip pat yra naudojami padėkliukai, pasikeitus prekės dydžiams taip pat turėtų būti keičiami ir jie. Verta paminėti, kad priklausomai nuo prekių svorio gali reikėti ir galingesnio magneto. Naudojant sugalvotą magnetui skirtą padėkliuką susidurta su bėda, kad priekyje esanti sienelė uždengia beveik visą prekės priekinę dalį, tai apsunkina kitų įrenginių tokių kaip barkodų skeneris darbą. Greičio atžvilgiu šis įrenginys taip pat turi tam tikrus limitus, priklausomai nuo prekių svorio bei magneto stiprumo. Atliktų bandymų metu didesniu greičiu traukiant prekę ji atsikabindavo nuo magneto. Taip pat vienas didžiausių šio metodo trūkumų yra papildomų padėkliukų poreikis. Apibendrinta įverčių lentelė pateikiama žemiau.

27 lentelė. Magnetinio paėmimo vertinimo rezultatas

	Tikslumas	Kaina	Pritaikomumas	Suderinamumas	Greitis	Papildoma įranga
Įvertis	3	3	3	3	3	2

Vakuuminis paėmimas

Vakuuminio paėmimo įrenginiu stengiamasi išspręsti problemas, kurias sukelia prieš tai buvusieji. Tikslumo šis būdas reikalauja nemažai, kadangi būtina pataikyti į tam tikroje vietoje stovinčią prekę, konkretus tikslumo poreikis priklauso nuo prekių, tačiau šiuo atveju tikslumas panašus kaip ir naudojant magnetą. Kainos atžvilgiu įrenginys yra identiškas variantui su magnetu, tik vietoj magneto įsigyjama vakuumo vožtuvas. Šis įrenginys yra lanksčiausias ir lengviausiai pritaikomas skirtingiems reikalavimams, nesunku keisti siurbtukus ar jų plotį, tiesa šio tipo įrenginys kol kas paima tik kietus ir lygius šonus turinčius daiktus. Verta pastebėti, kad sunkioms prekėms gali reikėti didesnių siurbtukų arba didesnio vakuumo srauto. Vakuumo paėmimo įranga netrikdo kitų įrenginių darbo.

Judėjimo greitis yra panašus su magnetinio įrenginio - per greitas prekės įtraukimas atkabina ją nuo siurbtukų. Didžiausias šio metodo privalumas – nereikalinga jokia papildoma įranga (padėkliukai).

28 lentelė. Vakuuminio paėmimo vertinimo rezultatas

	Tikslumas	Kaina	Pritaikomumas	Suderinamumas	Greitis	Papildoma įranga
Įvertis	3	3	4	5	3	5

Susumavus taškus gauti tokie rezultatai:

- Išsiskleidžiantis paėmimas - 20
- Magnetinis paėmimas - 17
- Vakuuminis paėmimas – 23

Taigi įvertinus ir išbandžius skirtingus paėmimo metodus nustatyta, kad geriausiai projekto poreikius atitinka vakuuminis paėmimas. Jis ir buvo pasirinktas bei naudojamas tolimesniuose tyrimuose.

5.4. Eksperimentų išvados

1. Pirmojo eksperimento metu išbandyti 3 skirtingi algoritmai, kurie sprendė celės centro koordinatinių apskaičiavimo paklaidų problemas naudojant lazerinį jutiklį. Kiekvieno algoritmo atveju skyrėsi procesas kaip yra apdorojamos po lentynų skenavimo gautos koordinatės. Geriausių rezultatų pavyko pasiekti su dažniausio pasikartojimo metodu. Taikant šį metodą pavyko sėkmingai įvažiuoti į visas 24 celes. Šis metodas pasiekė puikų rezultatą, nes X ašies paklaidos siekė tik 0.257mm, o Y ašies 0.559mm. Tai net 84% mažesnė X ašies ir 80% mažesnė Y ašies paklaida lyginant su pirmuoju sistemos kūrimo metu realizuotu metodu „Aritmetinis vidurkis“. Taigi dažniausio pasikartojimo metodas gali būti naudojamas sistemoje ir tinkamai atlieka vieną svarbiausių sistemos funkcijų.
2. Roboto stabdymo tikslumo pagerinimui naudojant DC tipo variklius taip pat buvo išbandyti 3 skirtingi metodai. Nors su lėtėjimo ir variklio blokavimo metodu pavyko sumažinti paklaidą nuo vidutiniškai 5,4cm iki 2,7cm, tačiau tokia paklaida vis tiek yra per didelė, kad robotas galėtų tiksliai atlikti reikiamas funkcijas. Taip pat buvo atlikti keli bandymai su BLDC tipo varikliais, kurie taip pat parodė neigiamus rezultatus. Po tokių rezultatų konsultavomės su technikos specialistais ir buvo nuspręsta, kad su turimais varikliais ir jų kontrolieriais nepavyks pasiekti geresnių rezultatų, tai yra daugiau tikslumo vien programinėmis pastangomis pasiekti nėra realu. Todėl šio tyrimo ir eksperimento metu išsiaiškinta, kad robotui reikia keisti variklius ir jų valdymą į žingsninio tipo (angl. stepper motor).
3. Trečiojo eksperimento metu išbandyti 3 skirtingi prekės paėmimo įrenginiai, kurie turi iš lentynos ištraukti prekes. Pagal sudarytą kriterijų lentelę įvertinus kiekvieną įrenginį nustatyta, kad geriausiai dalykinėje srityje funkcijas atlieka vakuuminis prekių paėmėjas. Vienas didžiausių šio įrenginio privalumas prieš kitus yra tas, kad nebereikia naudoti papildomų priedų norint ištraukti prekę. Kaip trūkumą pastebėjome, kad su šiuo tipu nepavyksta paimti nestandartinių formų arba pažeistų prekių, tačiau įvertinus sistemos siekiamą funkcionalumą tai nebuvo nulemiantis veiksnys

Išvados

1. Išanalizavus egzistuojančius logistikos automatizavimo sprendimus ir kliento keliamus reikalavimus bei turimos sistemos problemines sritis, nustatyta, kad rinkoje nėra produktų, kurie visiškai atitiktų keliamus reikalavimus ir kainos ribas.
2. Pasirėmus technologijų analize, įvertinus keliamus reikalavimus ir turimą biudžetą nuspręsta projektuoti automatiškai judantį sandėlio roboto prototipą, paremtą linijos sekimu ir žymių atpažinimu pozicijos nustatymui erdvėje.
3. Darbo metu suprojektuotas roboto prototipas, sukurta įrenginio ir jo komponentų valdymo sistema skirta reikiamoms funkcijoms vykdyti, taip išsprendžiant senojo produkto plečiamumo problemą. Realizuojant valdymo sistemą programuota C++ ir JavaScript programavimo kalbomis. Taip pat realizuota sąsaja komunikavimui su GRBL biblioteka, kuri valdo išorinį prekių paėmimo įrenginį.
4. Priėmimo testavimo metu pastebėta, kad trūksta tikslumo lentynos celės centro koordinatų nustatymo algoritme, roboto judėjime ir prekės paėmimo procedūroje. Tyrimo metu atlikti trys atskiri tyrimai kiekvienai problemai išanalizuoti, o tada atlikti eksperimentai patikrinti pasiūlytų sprendimų efektyvumą.
5. Pirmojo eksperimento metu išbandyti 3 skirtingi algoritmai, kurie sprendė celės centro koordinatų apskaičiavimo paklaidų problemas. Idealus rezultatas buvo pasiektas su tyrimo metu realizuotu dažniausio pasikartojimo metodu – tiksliai įvažiuota į visas 24 lentos celes. Metodus pasiekė tokį puikų rezultatą, nes X ašies paklaidos siekė tik 0.257mm, o Y ašies 0.559mm. Tai net 84% mažesnė X ašies ir 80% mažesnė Y ašies paklaida lyginant su pirmuoju sistemos kūrimo metu realizuotu metodu.
6. Antrojo eksperimento metu roboto sustojimo pagerinimui naudojant DC tipo variklius taip pat buvo išbandyti 3 skirtingi metodai. Taikant tyrimo metu realizuotą lėtėjimo ir variklio blokavimo metodą pavyko sumažinti paklaidą nuo vidutiniškai 5,4cm iki 2,7cm, tačiau tokia reikšmė vis tiek yra per didelė, kad robotas galėtų tiksliai atlikti reikiamas funkcijas. Eksperimentas parodė, kad vien programinėmis priemonėmis daug geresnių rezultatų pasiekti nepavyks, todėl būtina roboto variklius pakeisti į žingsninio tipo.
7. Trečiojo eksperimento metu nustatyta, kad geriausiai dalykiniai sričiai tinka vakuuminio paėmimo įrenginys. Nors šis paėmimo būdas sunkiai susidoroja su nestandartinių formų ar pažeistomis pakuotėmis, tačiau tai neturi didelės įtakos siekiamam sistemos taikymui. Vienas didžiausių šio įrenginio privalumų prieš kitus yra tas, kad nebereikia naudoti papildomų priedų (padėkliukų) norint ištraukti prekę. Šio tipo įrenginys bus tobulinamas toliau sistemos vystymo metu.
8. Užsakovas patenkintas gautu rezultatu, sistema atiduota ir įdiegta klientui (žr. 1 priedas.). Sukurta roboto valdymo sistema tinkamai atlieka savo funkcijas, o atliktas tyrimas ir eksperimentai parodė problemines sritis, kurias bus siekiama ištaisyti kuriant tobulesnę roboto versiją.

Literatūros sąrašas

1. Tarun Agarwal. „Robots – Types & Applications“ [žiūrėta 2017 m. gruodžio 2 d.]. Prieiga per internetą: <<https://www.elprocus.com/robots-types-applications/>>
2. Hohne, J.; Schreuder, M.; Blankertz, B.; Tangermann, M., "Two-dimensional auditory p300 speller with predictive text system," Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE , vol., no., pp.4185,4188, Aug. 31 2010-Sept. 4 2010
3. Matthew Edwards. „The Difference Between AGVs and Mobile Robots“ [žiūrėta 2017 m. gruodžio 2 d.]. Prieiga per internetą: <<https://www.linkedin.com/pulse/difference-between-agvs-mobile-robots-matthew-edwards/>>
4. Martí nez-Barbera´ H, Ca´novas JP,ZamoraM,Go´mez-Skarmeta A.iFork:a flexible AGV system using topological and grid maps. In: IEEE international conference on robotics and automation,Taipei,2003,pp.2147–2152.
5. Martí nez-Barbera´ H, Herrero-Perez D. „Autonomous navigation of an automated guided vehicle in industrial environments“ [žiūrėta 2017 m. gruodžio 2 d.]. Prieiga per internetą: <<http://www.sciencedirect.com/science/article/pii/S0736584509000994/>>
6. Gaskins, J. R.; Tanchoco A. M. J. „Flow path design for automated guided vehicle systems“ [žiūrėta 2017 m. gruodžio 2 d.]. Prieiga per internetą: <https://engineering.purdue.edu/~tanchoco/Papers/gaskins_tanchoco-IJPR-1987-v25n5.pdf/>
7. OTTO motors. „AGV vs SDV a comparison of automated material transport“ [žiūrėta 2017 m. gruodžio 2 d.]. Prieiga per internetą: <<https://www.ottomotors.com/resources/info/agv-vs-sdv>>
8. Chan C Y, Bougler B and Nelson D, “Characterization of magnetic tape and magnetic markers as a position sensing system for vehicle guidance and control”, Proc. Am. Control. Conf., Chicago, 2000, pp. 95-99.
9. Lu G and Tomizuka M, “Vehicle following as backup control schemes for magnetometer-based lateral guidance”, IEEE Trans. Control Syst. Technol., Vol. 13, No. 2, 2005, pp. 274-285.
10. FetchRobotics „Is Your Warehouse Ready for Autonomous Mobile Robots?“ [žiūrėta 2017 m. gruodžio 7.]. Prieiga per internetą: <<http://fetchrobotics.com/is-your-warehouse-ready-for-amrs/>>
11. Luca Carlone; Miguel Kaouk Ng; Jingjing Du; Basilio Bona; Marina Indri „Rao-Blackwellized Particle Filters Multi Robot SLAM with Unknown Initial Correspondences and Limited Communication“ [žiūrėta 2017 m. gruodžio 7 d.]. Prieiga per internetą: <https://areeweb.polito.it/ricerca/MacP4Log/index.php?option=com_content&view=article&id=25:rbwpmultirobotslamunkninitcorres&catid=9>
12. Xing, W., Peihuang, L., Ke, S., Guangqing, P., & Dunbing, T. (2014). PRECISE TRANSHIPMENT CONTROL OF AN AUTOMATED MAGNETIC-GUIDED VEHICLE USING OPTICS POSITIONING. *International Journal On Smart Sensing & Intelligent Systems*, 7(1), 48-71. Prieiga per internetą: <https://www.researchgate.net/publication/289214428_Precise_transshipment_control_of_an_automated_magnetic-guided_vehicle_using_optics_positioning>
13. Santos, P., Holé, S., Filloy, C., & Fournier, D. (2008). Magnetic vehicle guidance. *Sensor Review*, 28(2), 132-135. Prieiga per internetą: <<http://www.emeraldinsight.com/doi/abs/10.1108/02602280810856697>>

14. Vis Iris F A, "Survey of research in the design and control of automated guided vehicle systems", *Eur. J. Oper. Res.*, Vol. 170, No. 3, 2006, pp. 677-709.
15. Mariolino De Cecco. (2000). A new concept for triangulation measurement of AGV attitude and position. *Measurement Science and Technology*, 11(11), N105-N110. Prieiga per internetą: < <http://iopscience.iop.org/article/10.1088/0957-0233/11/11/401/meta> >
16. Fan, K., Yang, Q., Yan, W., & Lei, B. (2012). Fuzzy-PID based deviation-correcting control system for laser guided AGV. *Modelling, Identification & Control (ICMIC), 2012 Proceedings of International Conference on*, 472-477. Prieiga per internetą: < <http://ieeexplore.ieee.org.ezproxy.ktu.edu/document/6260279/> >
17. BORENSTEIN J., EVERETT H. R., FENG L., WEHE D. Mobile Robot Positioning: Sensors and Techniques. *Journal of Robotic Systems, Special Issue on Mobile Robots*, April, 1997, Vol. 14, No. 4. P. 231–249.
18. Fiodorova, O. (2016). Mobiliojo roboto koordinačių nustatymas naudojant žinomos aplinkos informaciją: Daktaro disertacija. Kaunas: Kauno technologijos universitetas. Prieiga per eLABa – nacionalinė Lietuvos akademinė elektroninė biblioteka.
19. KREJSA J., VECHET S. Infrared Beacons Based Localization of Mobile Robot. *Journal of Electronics and Electrical Engineering*, 2012, Vol. 117, No. 1. ISSN 1392-1215.
20. SHIMSHONI I. On Mobile Robot Localization from Landmark Bearings. *International Conference on Robotics and Automation*, 2001, Vol. 4. P. 3605–3611. ISBN 0-7803-6576-3.
21. SE S., LOWE D., LITTLE J. Mobile Robot Localization and Mapping with Uncertainty Using Scale-Invariant Visual Landmarks. *International Journal of Robotics Research*, 2002, Vol. 21, No. 8. P. 735–758.
22. ANGERMANN M. FootSLAM: Pedestrian Simultaneous Localization and Mapping without Exteroceptive Sensors Hitchhiking on Human Perception and Cognition. *Proceedings of the IEEE*, May, 2012, Vol. 100. P. 1840–1848.
23. BIBER P., ANDREASSON H., DUCKETT T., SCHILLING A. 3D Modelling of Indoor Environments by a Mobile Robot with a Laser Scanner and Panoramic Camera. *2004 International Conference on Intelligent Robots and Systems*, 2004, Vol. 4. P. 3430–3435. ISBN. 0-7803-8463-6.
24. ZIKOS N., PETRIDIS V. 6-Dof Low Dimensionality SLAM (L-SLAM). *Journal of Intelligent Robot System*, 2014, Vol. 79. P. 55–72.
25. ENGEL J., SCHOPS T., CREMERS D. LSD-SLAM: Large-Scale Direct Monocular SLAM. *European Conference on Computer Vision (ECCV)*, 2014. Vol. 8690, P. 834–849.
26. MUR-ARTAL R., MONTIEL M. M., TARDOS J. ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *Journal of Computer Vision and Recognition*, 2015. P. 1147–1163. ISBN 1552-3098.
27. BOURMAUD G., MEGRET R. Robust Large Scale Monocular Visual SLAM. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. P. 1638–1647.
28. LEE S., et al. DV-SLAM (Dual-Sensor-Based Vector-FieldSLAM) and Observability Analysis. *IEEE Transactions on Industrial Electronics*, February, 2015, Vol. 62, No. 2. P. 1101–1112. ISBN 0278-0046.
29. LEE S., SONG J.-B. Mobile Robot Localization Using Range Sensors: Consecutive Scanning and Cooperative Scanning. *International Journal of Control, Automation, and Systems*, March 2005, Vol. 3, No. 1. P. 1–14.

Priedai

1 priedas. Įdiegimo aktas

UAB „Aksmo sistemos“
Įmonės kodas: 304472994
Adresas: Versmės g. 3D-16, Jonučiai II, LT-53277 Kauno r.

PROGRAMŲ SISTEMOS PERDAVIMO IR APROBAVIMO AKTAS

201 9 m. sausio 15 d.

Programų sistemos pavadinimas Automatizuoto sandėlio roboto su vedliu projektas ir valdymo sistema

Kūrinio tipas Automatizuoto sandėlio roboto valdymo programinė įranga.

Programų sistemos sukūrimo data 201 9 m. sausio 4 d.

Kūrinio įteikimo UŽSAKOVUI data 201 9 m. sausio 8 d.

Užsakovo arba trečiojo asmens kūrinio aprobavimo rezultatas:

Programų sistemos „Automatizuoto sandėlio roboto valdymo programinė įranga“ paskirtis – valdyti automatizuoto AGV roboto veiklą ir techninę įrangą. Sukurta programinė įranga turi gebėti valdyti AGV tipo roboto techninę įrangą ir užduočių planavimo veiklą, naviguoti aplinkoje, kurioje yra linijomis nubrėžtas kelias ir kaip atskaitos taškai naudojamą RFID žymos bei komunikuoti su kitais įrenginiais: Valdymo sistema turi suskaičiuoti artimiausius kelius iki tikslo ir valdyti robotą jo važiavimo metu. Taip pat programinė įranga turi gebėti valdyti išorinį prekių paėmimo įrenginį, kuris valdomas naudojant GRBL biblioteką. Roboto komunikavimui su vartotoju realizuota API sąsaja. Tarp vidinių komponentų sistema komunikuoja apibrėžtomis vidinėmis žinutėmis. Sistema turi gebėti išsaugoti nustatymus ilgalaikėje aplinkoje bei rašyti sistemos veiksmus į failą. Mechaninio valdymo sistema realizuota programuojant Android mikrovaldiklį su C++ kalba. Užduočių planavimo sistema realizuota JavaScript kalba ir vykdoma NodeJs aplinkoje. Sistema įdiegta Android mikrokontroleryje ir RaspberryPi 3 kompiuteryje.

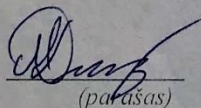
Kūrinio aprobavimo data 201 9 m. sausio 11 d.

Kūrinio originalo saugotojas – UAB „Aksmo sistemos“

AUTORIAI:

Mantas Dulinskas

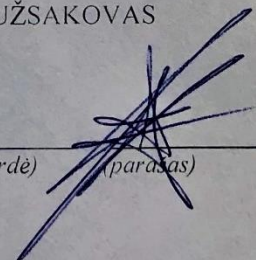
(vardas, pavardė)


(parašas)

UŽSAKOVAS

Andrius Kolesnikas

(vardas, pavardė)


(parašas)